

Apport de l'espace Teinte-Saturation-Luminance pour la segmentation spatiale et temporelle

Hue-Saturation-Luminance Colour Space for Spatial and Temporal Segmentation

Sébastien Lefèvre¹, Nicole Vincent²

¹LSIT – Université Louis Pasteur (Strasbourg 1) Parc d'Innovation, bd Brant, BP 10413, 67412 Illkirch cedex,
lefevre@lsiit.u-strasbg.fr

²CRIP5 – Université René Descartes (Paris 5) 45, rue des Saints Pères, 75270 Paris cedex 06,
nicole.vincent@math-info.univ-paris5.fr

Manuscrit reçu le 11 juillet 2005

Résumé et mots clés

Dans cet article, nous nous focalisons sur l'espace Teinte Saturation Luminance, réputé pour sa meilleure représentation de la vision humaine, et illustrons son intérêt dans la segmentation de données image, que ce soit dans l'espace ou dans le temps. L'espace TSL, aux caractéristiques particulières (une composante angulaire, la teinte, et deux composantes scalaires, la saturation et la luminance) nécessite des modes de calcul originaux. Notre contribution est donc triple : tout d'abord nous nous attarderons sur la représentation des couleurs dans l'espace TSL et sur les méthodes de calcul spécifiques qui doivent être employées. Ensuite nous montrerons l'intérêt de cet espace dans la segmentation automatique de données image dans les deux domaines que sont l'espace et le temps. La segmentation spatiale est assimilée au problème de la séparation du fond et des objets pour cela nous proposons une approche multirésolution ne nécessitant qu'une seule image. La segmentation temporelle correspond à la détection des changements de plan dans une séquence d'images, et la méthode que nous proposons pour l'obtenir se base sur l'utilisation de mesures de distances indépendamment du contexte. Les caractéristiques communes de nos deux méthodes (segmentation spatiale et segmentation temporelle) sont l'efficacité (temps de traitement permettant de respecter la cadence vidéo) et la robustesse (notamment aux changements d'illumination). Nous illustrons ces deux approches par des résultats obtenus dans le domaine de l'analyse de séquences vidéo sportives et comparons dans ce contexte l'usage des espaces TSL et RVB.

TSL, segmentation, multirésolution, illumination, séparation objets/fond, changements de plan.

Abstract and key words

In this article, we focus on the Hue Saturation Luminance colour space, known for its appropriate representation of the human vision, and we illustrate its interest in the segmentation of image data, both in the spatial and temporal dimensions. The HSL space, with its particular properties (one angular component, the hue, and two scalar components, the saturation and the luminance), requires some original computation models. So our contribution is triple : first we will deal with colour representation in the HSL space and with specific computation methods which have to be involved. Then we will show the relevance of this colour space for automatic segmentation of image data in the two domains which are space and time. Spatial segmentation is considered under the problem of background and foreground separation for which we propose a multiresolution approach which requires a single image. Temporal segmentation corresponds to shot change detection in an image sequence, and the method we are proposing is based on the use of

context-independent distance measures. Common properties of our both methods (spatial and temporal segmentations) are efficiency (processing time compatible with video framerate) and robustness (in particular against illumination changes). We illustrate these two approaches with results obtained in the domain of sport video sequences analysis and we compare in this context the use of HSL and RGB colour spaces.

HSL, segmentation, multiresolution, illumination, background / foreground separation, shot change.

1. Introduction

Historiquement, le traitement des images s'est longtemps limité à des images en niveaux de gris. Depuis une dizaine d'années, les avancées technologiques [8] ont permis de dépasser ce verrou et de considérer des images couleur. En effet, le développement de nouveaux capteurs trichromatiques a fourni les images couleur, tandis que l'amélioration des performances de calcul des processeurs a offert la possibilité d'accepter le surcoût de calcul inhérent à l'augmentation des données à traiter lors du passage des images en niveaux de gris aux images couleur. Cependant, la plupart des algorithmes d'analyse et de traitement d'images ou de séquences d'images reste limitée à des images en niveaux de gris et n'utilise pas toute la richesse d'information fournie par les images couleur. De plus, souvent les techniques proposées sous l'appellation « couleur » ne sont qu'une adaptation simpliste d'une technique pour les images en niveaux de gris à des images en couleur. Ainsi, l'espace couleur considéré est le plus souvent RVB (Rouge Vert Bleu) et le processus se limite à un traitement (prévu initialement pour une image en niveaux de gris) similaire pour les 3 composantes, suivi d'une étape de « fusion » (généralement une combinaison des résultats obtenus au sein de chaque image). Il est évident que l'intérêt de ce type d'approches reste limité et que les avancées en théorie de la couleur doivent être employées plus judicieusement pour obtenir des traitements robustes bénéficiant de manière approfondie de la richesse des informations couleur. Différents travaux recherchent à établir l'espace de travail le mieux adapté à une situation donnée. Si cette approche est envisageable dans le cas d'images fixes assez homogènes, à traiter en grand nombre, par contre changer d'espace au cours d'une séquence vidéo semble irréaliste.

Nous voulons montrer ici que l'utilisation de l'espace Teinte - Saturation - Luminance constitue un compromis remarquable qui permet d'améliorer les résultats obtenus dans l'espace Rouge - Vert - Bleu. Cet espace présente plusieurs avantages de natures différentes. Il assure une grande symétrie d'approche par rapport aux différentes teintes sans en privilégier certaines de manière arbitraire. De plus la nature des différents axes de représentation permet de diminuer la dimension de l'espace d'étude en fonction

du problème à résoudre, diminuant d'autant la complexité des algorithmes mis en œuvre. Cet espace permet également une meilleure adaptation aux différentes conditions d'acquisition des scènes. Néanmoins on doit noter que cet espace de représentation, notamment connu pour sa plus fidèle correspondance avec le système de vision humaine, présente certaines particularités (comme la présence de valeurs angulaires) qui nécessitent des outils de manipulation spécifiques. Nous nous proposons dans cet article d'illustrer ces différentes remarques au travers du problème de la segmentation, en considérant d'une part une segmentation spatiale dans les images fixes et d'autre part une segmentation temporelle dans les séquences vidéo. Plus précisément, nous considérerons successivement les domaines spatial et temporel et proposerons dans chacun de ces domaines une méthode de segmentation basée sur l'espace TSL. La segmentation spatiale sera assimilée à une séparation des objets et du fond, tandis que la segmentation temporelle consistera à détecter les changements de plan présents dans une séquence d'images. Notre contribution est donc triple : étude des outils nécessaires à la manipulation de données TSL, segmentation spatiale d'images dans TSL, et segmentation temporelle de séquences d'images dans TSL.

Notre contribution sera organisée selon le plan suivant. Nous présenterons tout d'abord l'espace TSL, ses caractéristiques et son obtention depuis l'espace RVB, nous décrirons notamment les modes de calcul qui lui sont spécifiques. Dans une seconde partie, nous aborderons le problème de la segmentation spatiale. Après avoir présenté succinctement le problème et rappelé quelques méthodes existantes, nous expliquerons notre approche en soulignant notamment l'intérêt d'un cadre multirésolution et son utilisation dans un espace couleur. Nous nous focaliserons ensuite sur la segmentation temporelle pour laquelle, après avoir revu les principales solutions de la littérature, nous proposons une méthode nécessitant un prétraitement des données et la définition d'une mesure de distance appropriée. Finalement nous présenterons et commenterons les résultats obtenus dans le contexte de l'analyse vidéo de séquences télévisées de matchs de football. Nous illustrerons ainsi l'intérêt de l'espace TSL par rapport à l'espace RVB, et la robustesse de nos approches.

2. Représentation et calculs dans l'espace TSL

Le codage de la couleur dans des images numériques peut être effectué en utilisant différents espaces de représentation, appelés traditionnellement espaces couleur. Pour plus d'informations, le lecteur pourra se référer au récent ouvrage de Trémeau *et al.* [19] ou au panorama de Chang *et al.* [3] sur la segmentation couleur. Dans cet article, nous nous focalisons sur l'espace TSL. Nous expliquerons donc ici la représentation des couleurs dans l'espace TSL et donnerons les formules de conversion depuis l'espace RVB. Puis nous rappellerons et introduirons des méthodes appropriées au calcul et à la manipulation des données dans cet espace.

2.1. Représentation et conversion

L'espace TSL comprend trois composantes : la teinte, la saturation, et la luminance. Tandis que la saturation et la luminance sont codées « de manière classique » sous forme scalaire, la teinte est pour sa part une valeur angulaire. Ces composantes peuvent être interprétées de la manière suivante :

Teinte : représente la couleur perçue (rouge, jaune, vert, etc.),

Saturation : mesure la pureté de la couleur (par exemple pour une teinte rouge, le rose se caractérise par une saturation plus faible que le rouge, tandis que le noir, le blanc et le gris sont caractérisés par une saturation nulle),

Luminance : représente le niveau de gris, de « sombre » pour une valeur faible à « clair » pour une valeur élevée.

La figure 1 illustre ces différents concepts. Nous pouvons observer que la saturation et la luminance sont toutes deux représentées sur une échelle linéaire tandis que la teinte est modélisée

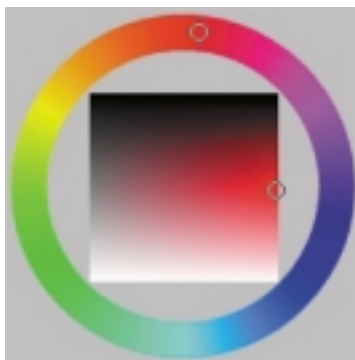


Figure 1. Représentation visuelle de l'espace TSL : le disque modélise la teinte, tandis que le carré modélise la saturation (direction horizontale) et la luminance (direction verticale).

par un angle. Elle sera donc logiquement l'objet d'une considération particulière lorsqu'elle sera employée dans des calculs variés, ce que nous verrons plus loin dans cette section.

L'espace TSL fournit, au travers de ses 3 composantes, des informations complémentaires. Il a l'avantage de permettre l'élaboration de méthodes robustes aux changements d'illumination. En effet, ces artefacts affectent principalement la composante de luminance. En ne tenant compte que des composantes de chrominance (teinte et saturation), il est donc possible de diminuer la sensibilité aux changements d'illumination.

Nous avons également observé que la teinte était une composante plus robuste que la saturation ou la luminance dans un cadre multirésolution, comme par exemple pour la méthode de segmentation spatiale que nous présenterons en section 3. En effet, la teinte est moins sensible aux artefacts dus à des moyennages successifs des valeurs des pixels, nécessaires dans le cas d'une représentation multirésolution pyramidale. Nous avons ainsi observé une indépendance vis-à-vis de la résolution de certaines mesures calculées sur les teintes (comme la plage de valeurs ou l'écart-type).

La teinte est donc une composante intéressante, invariante aux changements d'illumination et aux cadres multirésolutions. Cependant elle doit être analysée avec précaution. En effet, sa fiabilité dépend du niveau de saturation et la teinte n'est significative que si la saturation est élevée. Les méthodes d'analyse basées sur la teinte des pixels doivent donc vérifier que ceux-ci ne sont pas achromatiques. Une autre contrainte de la teinte provient de sa nature mathématique (mesure angulaire) qui nécessite l'utilisation de mesures statistiques spécifiques décrites par la suite.

À partir des concepts généraux que nous venons de rappeler, différents auteurs ont proposé leur propre modèle TSL. Ainsi, on peut citer les modèles décrits dans les ouvrages de Travis [18] ou de Gonzalez et Woods [7] ou le modèle introduit plus récemment par Serra dans [17]. Pour un panorama étendu, le lecteur pourra se référer aux travaux de Carron [2] ou de Vandembroucke [21]. La grande variabilité des modèles TSL introduits entraîne nécessairement des formules de conversion différentes. Nous utiliserons ici la définition de Travis [18] que nous avons préférée à celle de Gonzalez et Woods [7]. En effet, dans cette dernière approche, la conversion depuis RVB vers TSL s'accompagne d'un coût de calcul plus élevé, pour des résultats équivalents. Le modèle que nous avons choisi est également identifié sous le terme *modèle de cône hexagonal*, il est décrit dans [21].

Avant de définir précisément l'espace TSL, nous introduisons la notation $I(x, y, C)$ pour désigner la valeur du pixel de coordonnées (x, y) pour la caractéristique couleur C . Ainsi sont définis :

$$I(x, y, \min_{RVB}) = \min(I(x, y, R), I(x, y, V), I(x, y, B)) \quad (1)$$

$$I(x, y, \max_{RVB}) = \max(I(x, y, R), I(x, y, V), I(x, y, B)) \quad (2)$$

Les coordonnées dans l'espace TSL s'expriment alors en fonction des coordonnées dans l'espace RVB, pour la luminance ;

elles sont données par :

$$I(x, y, L) = I(x, y, \max_{RVB}) \quad (3)$$

L'inconvénient de cette définition réside dans la sensibilité au bruit de la composante de luminance. Néanmoins, comme nous pourrions le constater par la suite, les méthodes de segmentation que nous proposons se basent essentiellement sur les composantes de teinte et de saturation et de ce fait s'affranchissent en partie de cette contrainte. Si $I(x, y, L)$ est nul, la saturation et la teinte sont indéfinies. Dans le cas contraire, on a :

$$I(x, y, L) \neq 0 \text{ et } I(x, y, S) = \frac{I(x, y, L) - I(x, y, \min_{RVB})}{I(x, y, L)} \quad (4)$$

Si $I(x, y, S)$ est nul, la teinte est indéfinie. Sinon, la teinte est calculée en radians comme :

$$I(x, y, T) = \begin{cases} \frac{\pi}{3}(5 + I(x, y, B')) & \text{si } I(x, y, R) \geq I(x, y, B) \geq I(x, y, V) \\ \frac{\pi}{3}(1 - I(x, y, V')) & \text{si } I(x, y, R) \geq I(x, y, V) \geq I(x, y, B) \\ \frac{\pi}{3}(1 + I(x, y, R')) & \text{si } I(x, y, V) \geq I(x, y, R) \geq I(x, y, B) \\ \frac{\pi}{3}(3 - I(x, y, B')) & \text{si } I(x, y, V) \geq I(x, y, B) \geq I(x, y, R) \\ \frac{\pi}{3}(3 + I(x, y, V')) & \text{si } I(x, y, B) \geq I(x, y, V) \geq I(x, y, R) \\ \frac{\pi}{3}(5 - I(x, y, R')) & \text{si } I(x, y, B) \geq I(x, y, R) \geq I(x, y, V) \end{cases} \quad (5)$$

avec $I(x, y, R')$, $I(x, y, V')$, $I(x, y, B')$ calculés par l'équation suivante :

$$I(x, y, c') = \frac{I(x, y, \max_{RVB}) - I(x, y, c)}{I(x, y, \max_{RVB}) - I(x, y, \min_{RVB})} \quad (6)$$

$\forall c \in \{R, V, B\}$

qui peut également s'écrire :

$$I(x, y, c') = \frac{I(x, y, L) - I(x, y, c)}{I(x, y, S) \times I(x, y, L)} \quad \forall c \in \{R, V, B\} \quad (7)$$

Nous avons décrit ici la signification des trois composantes de l'espace TSL et le moyen d'obtenir une représentation dans cet espace à partir de données RVB. Nous avons rappelé comment il était possible de garantir la robustesse à des changements d'illumination, et l'intérêt apporté par la composante de teinte, tout en soulignant sa nature angulaire. Nous allons maintenant décrire quelques modes de calcul adaptés à ce type de données.

2.2. Méthodes de calcul

Nous avons mentionné précédemment la nature angulaire de la teinte. Cette composante devra donc être traitée de manière appropriée et des méthodes de calcul adaptées aux valeurs angulaires doivent être utilisées. Nous présentons ici quelques techniques permettant de manipuler correctement des données angulaires comme la teinte.

Lorsqu'on cherche à mesurer la similarité entre deux valeurs, on est logiquement amené à calculer la différence absolue entre ces deux valeurs. Dans le cas de valeurs angulaires, le calcul doit être fait modulo 2π , ce qui conduit à la formule suivante :

$$|\theta_i - \theta_j|_{\mathcal{L}} = \min(|\theta_i - \theta_j|, 2\pi - |\theta_i - \theta_j|) \quad (8)$$

en notant $|\theta_i - \theta_j|_{\mathcal{L}}$ la différence absolue entre deux valeurs angulaires θ_i et θ_j .

En cas de fusion ou de combinaison d'informations, la moyenne est fréquemment utilisée en segmentation des images. Nous avons choisi d'utiliser la définition donnée dans [12] et reprise dans [4] pour le calcul de la moyenne $\bar{\theta}$ d'un ensemble de mesures d'angle $\{\theta_i\}_{i \in [1, \Theta]}$. Elle s'obtient par la formule :

$$\bar{\theta} = \begin{cases} \text{Arctan} \left(\frac{\sum_{i=1}^{\Theta} \sin(\theta_i)}{\sum_{i=1}^{\Theta} \cos(\theta_i)} \right) & \text{si } \sum_{i=1}^{\Theta} \cos(\theta_i) \geq 0, \\ \text{Arctan} \left(\frac{\sum_{i=1}^{\Theta} \sin(\theta_i)}{\sum_{i=1}^{\Theta} \cos(\theta_i)} \right) + \pi & \text{sinon} \end{cases} \quad (9)$$

Dans [12, 4] est donné également un algorithme permettant le calcul de l'amplitude de variation de l'ensemble $\{\theta_i\}_{i \in [1, \Theta]}$. Cette méthode nécessite un tri croissant préalable de tous les angles considérés, et est donc caractérisée par une complexité algorithmique relativement élevée. Nous proposons ici une méthode applicable dans le cas où l'angle moyen $\bar{\theta}$ a déjà été calculé. Cette méthode nécessite, en plus de la moyenne, la connaissance des angles minimum et maximum dans l'intervalle de longueur 2π choisi, respectivement notés θ_{\min} et θ_{\max} . Si la moyenne appartient à l'intervalle limité par les deux angles, c'est-à-dire si $\theta_{\min} \leq \bar{\theta} \leq \theta_{\max}$, alors l'amplitude de variation est égale à $\theta_{\max} - \theta_{\min}$. Dans le cas contraire, l'angle complémentaire doit être considéré et l'amplitude de variation est égale à $2\pi - (\theta_{\max} - \theta_{\min})$. Si on a choisi des mesures d'angle comprises entre 0 et 2π , l'amplitude de variation $A(\{\theta_i\}_{i \in [1, \Theta]})$ est définie par :

$$A(\{\theta_i\}_{i \in [1, \Theta]}) = \begin{cases} \max_{i \in [1, \Theta]}(\theta_i) - \min_{i \in [1, \Theta]}(\theta_i) & \text{si } \min_{i \in [1, \Theta]}(\theta_i) \leq \bar{\theta} \leq \max_{i \in [1, \Theta]}(\theta_i), \\ 2\pi - (\max_{i \in [1, \Theta]}(\theta_i) - \min_{i \in [1, \Theta]}(\theta_i)) & \text{sinon} \end{cases} \quad (10)$$

Par la suite, nous utiliserons les différentes définitions introduites ici pour mesurer et comparer des valeurs angulaires, la teinte dans notre contexte. Nous avons décrit ici l'espace TSL et ses caractéristiques. Nous allons maintenant justifier son intérêt par l'apport de deux méthodes de segmentation, l'une dans le domaine spatial et l'autre dans le domaine temporel.

3. Segmentation spatiale

Nous avons précédemment décrit l'espace TSL qui nous semble opportun pour le traitement d'images couleur. Afin d'illustrer son intérêt, nous aborderons le problème de la segmentation sous deux aspects : dans le domaine spatial et dans le domaine temporel. Nous avons délibérément choisi d'assimiler le problème de la segmentation spatiale, que nous allons traiter ici, à celui de la séparation des objets et du fond. Après avoir rappelé les caractéristiques de ce problème, nous rappellerons les principales méthodes existantes pour résoudre ce problème et soulignerons leurs limites. Puis nous introduirons une solution originale basée sur l'utilisation de l'espace TSL dans un cadre multi-résolution.

3.1. Description du problème

Nous avons choisi de considérer le problème de la segmentation spatiale en tant que séparation du fond et des objets. Ce choix nous semble opportun dans la mesure où la séparation fond/objets est cruciale dans de nombreuses applications, telles que le suivi d'objet, l'interprétation du contenu des images et des séquences d'images, ou encore la compression. En effet, la norme de compression MPEG-4 décrit une scène par les différents objets qui la composent et par son arrière-plan [9].

Le domaine (noté \mathcal{I}) de définition des images d'une séquence vidéo peut être décomposé la plupart du temps en deux parties complémentaires : la scène ou l'environnement qui est représenté par l'arrière-plan, et les objets mobiles présents dans la scène. En notant \mathcal{R} le domaine occupé par les objets, nous pouvons définir $\overline{\mathcal{R}} = \mathcal{I} \setminus \mathcal{R}$ le domaine occupé par l'arrière-plan de la scène. La détection des objets et celle de l'arrière-plan sont donc deux problèmes duaux et il est possible de déterminer les pixels de l'image appartenant à l'une de ces deux parties par recherche des pixels de la partie complémentaire. Dans le cadre de la détection d'événements dans des séquences vidéo, les images analysées présentent souvent un partitionnement non équitable entre l'arrière-plan et des objets. En effet, l'arrière-plan occupe fréquemment une partie du champ de vision plus importante que les objets en mouvement présents dans la scène, ce qui se traduit par $\text{aire}(\mathcal{R}) \ll \text{aire}(\overline{\mathcal{R}})$. Cela permet une meilleure interprétation des actions effectuées par les différents objets dans leur environnement.

Nous aborderons donc ici le problème de la séparation du fond et des objets dans ce type d'images. Les objets considérés ne doivent pas être prépondérants dans l'image. Nous ne traiterons pas par exemple le cas des images de visage où le visage occupe quasiment toute l'image. En effet, notre objectif n'est pas ici de proposer une méthode de segmentation générique mais de montrer l'intérêt de l'espace couleur TSL en traitement des images, au travers de différentes applications. Néanmoins nous tiendrons compte des changements d'illumination et du temps de calcul nécessaire.

Le problème de la séparation du fond et des objets fait l'objet de différentes méthodes et nous allons rappeler les plus classiques, adaptées à des séquences acquises soit avec une caméra fixe (statique), soit en mouvement (dynamique).

3.2. Description de quelques solutions classiques

Nous présentons les techniques les plus couramment utilisées en considérant tout d'abord le cas d'une caméra statique, puis celui d'une caméra dynamique. Les principales limites soulignées nous permettront de justifier l'intérêt d'une nouvelle approche.

Dans le cas où la caméra effectuant l'acquisition est fixe, la séparation du fond et des objets peut être obtenue par différentes techniques de faible complexité algorithmique. Ces techniques sont basées sur la comparaison des images successives de la séquence. Plus précisément, la cohérence de l'intensité ou de la couleur d'un pixel au cours du temps influe directement sur la classification qu'on lui donne : fond ou objet. Ainsi, une première approche suppose que l'arrière-plan est connu initialement *via* la disponibilité d'une image de référence I_{ref} . Cette image représente uniquement la scène, aucun objet n'étant présent. Pour une image donnée I_t , la classification d'un pixel $P_t(x, y)$ en objet ou en fond résulte de la comparaison avec l'image de référence :

$$P_t(x, y) \text{ classé en } \begin{cases} \text{objet} & \text{si } |I_t(x, y) - I_{\text{ref}}(x, y)| > S, \\ \text{fond} & \text{sinon} \end{cases} \quad (11)$$

S étant un seuil de tolérance à fixer empiriquement ou après apprentissage [15, 16]. Cependant, il est possible dans certains cas qu'aucune image de référence ne soit disponible. La séparation des objets et du fond s'effectue alors par comparaison d'images successives, en considérant un intervalle de temps Δ_t plus ou moins important. La classification d'un pixel est alors donnée par :

$$P_t(x, y) \text{ classé en } \begin{cases} \text{objet} & \text{si } |I_t(x, y) - I_{t-\Delta_t}(x, y)| > S, \\ \text{fond} & \text{sinon} \end{cases} \quad (12)$$

D'autres méthodes plus robustes aux changements d'illumination ont également été proposées.

Dans le cas d'une caméra mobile, les méthodes précédentes ne sont plus applicables directement. La méthode employée est généralement composée de trois étapes successives : estimation du mouvement de la caméra, compensation de ce mouvement, et utilisation d'une approche pour caméra statique. Tout d'abord, il est nécessaire d'estimer le mouvement de la caméra, ce qui revient à estimer le mouvement global observé dans la séquence. De nombreuses méthodes permettent de résoudre ce problème. On pourra par exemple se référer aux ouvrages de Tziritas et Labit [20] ou plus récemment de Wang *et al.* [23]. Une fois le mouvement de la caméra estimé, une technique de compensation du mouvement est utilisée [5], permettant d'annuler dans

l'image les effets du mouvement de la caméra. Les images obtenues peuvent donc être considérées comme acquises avec une caméra fixe [1]. La dernière étape consiste alors en l'utilisation d'une méthode pour caméra fixe, telle que celles présentées précédemment. Là encore, la qualité du résultat peut être améliorée en utilisant un modèle plus robuste ou en ajoutant des post-traitements.

Ces deux types de méthodes présentent des limites importantes. Tandis que les premières sont incapables de gérer le cas d'une caméra en mouvement, les secondes nécessitent un temps de calcul important (pour l'estimation et la compensation du mouvement). De plus, on peut remarquer que dans ces méthodes la couleur n'a pas modifié le principe élaboré *a priori* sur les images en niveaux de gris. Les différences scalaires ont été remplacées par des différences vectorielles dans l'espace RVB. Partant de cette constatation, nous proposons une méthode basée sur l'analyse d'une seule image et qui ne tient pas compte de l'information de mouvement puisque celle-ci ne peut être obtenue et traitée rapidement. En outre, la méthode que nous allons présenter est appropriée au cas des images fixes, où le modèle de l'arrière-plan ne peut être amélioré au cours du temps et où aucune information de mouvement n'est disponible. En contrepartie, elle suppose des scènes où l'arrière-plan occupe une partie du champ de vision plus importante que les objets (statiques ou en mouvement), comme nous le verrons dans les prochains paragraphes.

3.3. Intérêt d'une approche multirésolution

Comme énoncé précédemment, nous proposons une solution pour résoudre le problème de la séparation des objets et du fond en se basant sur l'analyse d'une seule image dans l'espace TSL. Nous rappelons que nous nous focalisons sur le cas simultané d'un fond homogène, prédominant dans l'image, principalement chromatique, et d'une caméra mobile dont les paramètres (intrinsèques ou extrinsèques) ne sont pas connus. Dans ce contexte, plusieurs solutions peuvent être envisagées pour estimer la couleur principale de l'arrière-plan. Puisque l'arrière-plan n'est généralement pas composé d'une couleur unique mais plutôt de plusieurs couleurs proches, la recherche du mode significatif de l'histogramme des couleurs nous semble délicate. Nous lui préférons donc une approche par filtrage des détails où l'objectif est de ne conserver que les couleurs correspondant à l'arrière-plan. Différentes techniques peuvent être employées dans cet objectif, parmi lesquelles la morphologie mathématique ou l'analyse multirésolution. C'est sur cette dernière solution que nos efforts se sont portés.

Nous allons donc justifier ici notre choix d'un cadre multirésolution. Pour cela, considérons que l'image I représentant la scène est étudiée par un observateur donné. On peut remarquer que la distance d entre l'observateur et l'image influe directement sur sa capacité à visualiser les détails présents dans l'image. La figure 2 illustre ce principe. Si l'observateur est suffi-

samment éloigné de l'image, il ne distingue dans celle-ci que l'arrière-plan, puisque nous considérons toujours que l'arrière-plan occupe la partie la plus importante de l'image, comparativement aux objets présents dans la scène. Plus l'observateur est proche de l'image, plus il est capable de distinguer les objets de manière précise.

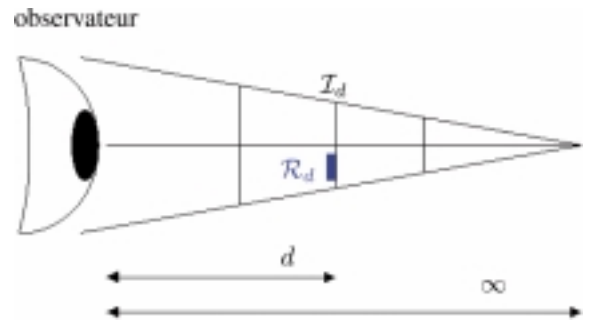


Figure 2. Vue d'une image par un observateur fixe à différentes distances d .

Le constat énoncé précédemment peut être modélisé de manière plus formelle. Pour cela, conservons les notations \mathcal{I} , \mathcal{R} , et $\overline{\mathcal{R}}$ représentant les domaines respectivement occupés par l'image, les objets et la scène. Ceux-ci dépendent de la distance d et seront donc notés plus précisément \mathcal{I}_d , \mathcal{R}_d , et $\overline{\mathcal{R}}_d$. Affinons de plus la définition de \mathcal{R}_d par la propriété suivante. Quand une région \mathcal{R}_d a une aire inférieure à ε , on redéfinit \mathcal{R}_d comme l'ensemble \emptyset . ε représente donc l'aire en dessous de laquelle une région ne peut plus être distinguée par l'observateur humain. D'après les définitions de \mathcal{R} et $\overline{\mathcal{R}}$, nous avons :

$$\mathcal{I}_d = \mathcal{R}_d \cup \overline{\mathcal{R}}_d \quad \forall d \in \mathbb{R}^+ \tag{13}$$

soit :

$$\text{aire}(\mathcal{I}_d) = \text{aire}(\mathcal{R}_d) + \text{aire}(\overline{\mathcal{R}}_d) \quad \forall d \in \mathbb{R}^+ \tag{14}$$

et :

$$\text{aire}(\mathcal{R}_d) \ll \text{aire}(\overline{\mathcal{R}}_d) \quad \forall d \in \mathbb{R}^+ \tag{15}$$

En considérant les deux équations précédentes, et en passant à la limite, nous obtenons :

$$\lim_{d \rightarrow \infty} \text{aire}(\mathcal{R}_d) + \lim_{d \rightarrow \infty} \text{aire}(\overline{\mathcal{R}}_d) = \lim_{d \rightarrow \infty} \text{aire}(\mathcal{I}_d) \tag{16}$$

$$\lim_{d \rightarrow \infty} \frac{\text{aire}(\mathcal{R}_d)}{\text{aire}(\mathcal{I}_d)} + \lim_{d \rightarrow \infty} \frac{\text{aire}(\overline{\mathcal{R}}_d)}{\text{aire}(\mathcal{I}_d)} = 1 \tag{17}$$

En considérant les propriétés formulées précédemment, nous avons alors :

$$\lim_{d \rightarrow \infty} \frac{\text{aire}(\mathcal{R}_d)}{\text{aire}(\mathcal{I}_d)} = 0 \quad (18)$$

et

$$\lim_{d \rightarrow \infty} \frac{\text{aire}(\overline{\mathcal{R}}_d)}{\text{aire}(\mathcal{I}_d)} = 1 \quad (19)$$

En effet, nous pouvons affirmer que :

$$\exists d_0 \quad / \quad \forall d > d_0 \quad \mathcal{R}_d = \emptyset \quad \text{et} \quad \overline{\mathcal{R}}_d = \mathcal{I}_d \quad (20)$$

Cette réflexion nous amène à proposer ici une approche multi-résolution. En considérant l'image originale $I_{(0)}$, il est possible de diminuer fortement sa résolution pour obtenir une image à très faible résolution $I_{(r_{\max})}$. Aucun objet n'est plus perceptible dans cette image qui n'est composée que de l'arrière-plan. Un modèle du fond peut donc être calculé à partir de cette image. En augmentant la résolution r de manière itérative, il est alors possible de comparer les différentes régions de l'image $I_{(r)}$ avec le modèle de l'arrière-plan suivant un critère donné. Cette comparaison permet de déterminer si chaque région correspond ou non à l'arrière-plan. De plus, le principe de multirésolution permet d'analyser des données à un degré de précision plus ou moins fin selon le résultat attendu. Une analyse multirésolution a aussi généralement l'avantage de limiter le nombre de calculs nécessaires par rapport à son homologue pleine résolution [14]. L'inconvénient principal est lié à la difficulté du choix des résolutions initiale et finale de l'analyse.

Nous allons maintenant décrire plus en détail la méthode multi-résolution de segmentation spatiale que nous proposons.

3.4. Méthode de segmentation

La séparation du fond et des objets s'effectue en quatre étapes successives : création de la représentation multirésolution, estimation du modèle de l'arrière-plan, segmentation itérative des images aux différentes résolutions et enfin segmentation finale. La première étape consiste en la création de la représentation multirésolution de l'image, qui peut être obtenue de différentes manières. Ainsi, une transformée en ondelettes peut être utilisée [22], mais elle nécessite généralement un temps de calcul important. Nous lui avons donc préféré une décomposition pyramidale [10]. La décomposition pyramidale d'une image est illustrée dans la figure 3. Afin de calculer la valeur d'un pixel $P(x, y)$ à la résolution $r + 1$ à partir d'un ensemble de p^2 pixels à la résolution r , nous utilisons la valeur moyenne des pixels de l'ensemble en question. La taille de l'image dépend alors de la résolution, et on a $X_{(r)} = \frac{X}{p^r}$ et $Y_{(r)} = \frac{Y}{p^r}$. La moyenne a été préférée à d'autres mesures nécessitant des calculs plus importants, comme par exemple la valeur médiane. Le calcul est effectué itérativement, à partir de la résolution originale $r = 0$, et jusqu'à obtenir la résolution voulue $r = r_{\max}$.

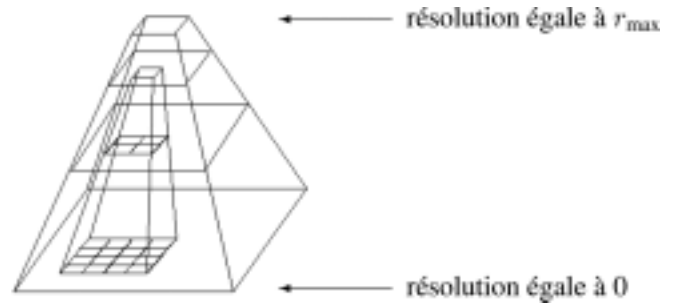


Figure 3. Représentation pyramidale d'une image.

Il est alors possible de modéliser l'arrière-plan avant de réaliser le processus de segmentation. Comme il a été précisé auparavant, nous considérons que l'arrière-plan peut être modélisé par l'image $I_{(r_{\max})}$, sommet de la pyramide. Cette considération n'est bien sûr valable que si l'arrière-plan occupe une partie importante de l'image, s'il diffère suffisamment des objets présents dans la scène, et s'il est assez homogène. Afin de garantir une faible complexité tout en assurant une robustesse aux changements d'illumination, nous avons décidé d'utiliser comme modèle la moyenne des teintes d'une zone, puisque la teinte nous semblait une composante relativement robuste aux moyennages successifs nécessaires pour construire la représentation multirésolution de l'image. Le modèle de l'image I , noté $\varphi(I)$, est donc défini comme :

$$\varphi(I) = \begin{cases} \text{Arctan} \left(\frac{\sum_{x=1}^X \sum_{y=1}^Y \sin(I(x, y, T))}{\sum_{x=1}^X \sum_{y=1}^Y \cos(I(x, y, T))} \right) \\ \quad \text{si } \sum_{x=1}^X \sum_{y=1}^Y \cos(I(x, y, T)) \geq 0, \\ \text{Arctan} \left(\frac{\sum_{x=1}^X \sum_{y=1}^Y \sin(I(x, y, T))}{\sum_{x=1}^X \sum_{y=1}^Y \cos(I(x, y, T))} \right) + \pi \quad \text{sinon} \end{cases} \quad (21)$$

L'arrière-plan sera donc caractérisé par $\varphi(I_{(r_{\max})})$. Ce choix requiert la validité de deux hypothèses. D'une part, il est nécessaire que les pixels ne soient pas achromatiques (saturation non nulle) pour que la valeur de leur teinte soit fiable. Dans notre approche, nous ne considérons donc que les pixels de saturation supérieure à un seuil prédéfini S_S , et nous nous assurons que leur proportion est suffisante pour garantir une certaine validité statistique. D'autre part, l'arrière-plan doit être de couleur (ou plutôt de teinte) homogène. Dans cet article, notre but est de montrer l'intérêt de l'espace TSL pour la segmentation d'images et nous ne considérons donc pas ici ces hypothèses comme des limites.

La segmentation peut ensuite être effectuée et améliorée de manière itérative (à la manière d'un *quadtree* incomplet) depuis la résolution $r_{\max} - 1$ jusqu'à la résolution initiale $r = 0$, base de la pyramide. À une résolution donnée r' avec $r_{\max} > r' > 0$, l'image $I_{(r')}$ doit être analysée. Cette image est comparable à l'image initiale $I_{(0)}$ qui aurait été découpée en $K = (K_0)^{r_{\max} - r'}$ zones, avec K_0 une constante dont la valeur pourrait être en toute logique égale à p^2 . Chacune de ces zones est alors comparée avec le modèle de l'arrière-plan. Cet appariement entre deux zones I^m et I^n s'effectue en calculant la mesure de similarité δ sur les moyennes respectives des teintes :

$$\delta(\varphi(I^m), \varphi(I^n)) = \min\left(\frac{|\varphi(I^m) - \varphi(I^n)|}{2\pi - |\varphi(I^m) - \varphi(I^n)|}\right) \quad (22)$$

Une fois la mesure δ calculée entre le modèle d'une zone donnée et le modèle de l'arrière-plan, elle est comparée à un seuil S_1 afin d'effectuer ou non la reconnaissance. Une valeur inférieure au seuil signifie que la zone est considérée comme l'arrière-plan.

Cependant, un second test est nécessaire pour vérifier la cohérence de la zone étudiée et éviter les artefacts liés à l'utilisation de la moyenne. En effet, si une zone est composée de deux pixels ayant des teintes opposées, la moyenne ne reflétera pas correctement le contenu de la zone. Nous analysons donc la cohérence de chaque zone appariée avec l'arrière-plan. Pour cela, plusieurs mesures de dispersion sont disponibles, comme la variance ou l'amplitude de variation. C'est cette dernière mesure qui est utilisée pour évaluer ici la cohérence d'une zone : plus la plage de valeurs d'une zone est faible, plus celle-ci est homogène. Pour calculer cette plage de valeurs angulaires, nous n'utilisons pas la méthode donnée dans [12, 4] mais l'approche originale présentée précédemment. Une fois la plage de valeurs calculée pour une zone $I_{(r)}^k$ par l'équation (10), nous comparons cette mesure de dispersion avec un second seuil S_2 . Une plage inférieure au seuil assure l'homogénéité de la zone concernée. Celle-ci est alors étiquetée en fond ou arrière-plan. Dans le cas contraire, l'hétérogénéité de la zone candidate à l'étiquetage implique son rejet.

Si une zone fournit une réponse positive à ces deux tests successifs, elle est affectée à l'arrière-plan. Dans ce cas la zone n'est plus analysée à de meilleures résolutions. À l'opposé, une zone sans étiquette sera analysée plus en détail à la résolution $r' - 1$. Cette segmentation est effectuée si nécessaire jusqu'à la résolution initiale $r = 0$. Dans le cas d'applications nécessitant une segmentation très précise, les zones correspondant aux objets peuvent être analysées par la suite afin d'affiner les contours des objets. Au contraire, si la précision des contours des objets n'est pas nécessaire (comme dans le cas de l'initialisation d'un algorithme de suivi d'objet), le processus peut être arrêté à une résolution r_{final} avec $r_{\max} > r_{\text{final}} \geq 0$. Dans ce cas, nous considérons que les zones sans étiquette à la résolution courante r_{final} représentent les objets.

Afin d'améliorer la qualité du modèle de l'arrière-plan, il est possible de recalculer celui-ci au cours du processus de seg-

mentation. Dans ce cas, le modèle obtenu à la résolution r_{\max} ne représente que l'état initial de l'arrière-plan. À mesure que la résolution devient plus fine, les résultats sont plus précis, et il est possible d'obtenir un modèle de l'arrière-plan plus fiable en ne considérant que les parties de l'image déjà étiquetées en fond. En effet, supposons que le processus a été réalisé depuis la résolution r_{\max} où le modèle de l'arrière-plan a été estimé jusqu'à la résolution r' . Chaque pixel à la résolution r' a été étiqueté comme fond ou objet. Le modèle de l'arrière-plan peut donc être amélioré, c'est pourquoi nous le qualifions d'évolutif. Le nouveau modèle obtenu peut alors être utilisé pour étiqueter les pixels de l'image à la résolution $r' - 1$.

Nous avons décrit ici comment l'espace TSL pouvait être utilisé pour obtenir une méthode de segmentation spatiale robuste et efficace. Nous allons montrer maintenant qu'il peut également être employé dans le cas d'une segmentation des données non plus spatiale mais temporelle.

4. Segmentation temporelle

Après avoir abordé le problème de la segmentation spatiale assimilée ici à la séparation des objets et du fond, et proposé une solution basée sur l'espace TSL, nous allons étudier maintenant le problème de la segmentation temporelle, ou détection des changements de plan. Nous commencerons par définir les différents types de changements de plan et rappellerons les principales familles de méthodes pour résoudre ce problème. Nous présenterons ensuite notre approche en détaillant le nécessaire prétraitement des données, la définition de la mesure de distance utilisée, et enfin la solution globale proposée.

4.1. Présentation du problème

Un plan est défini comme une suite d'images issues d'une acquisition continue d'une caméra donnée. Ainsi, toutes les images d'un plan ont été acquises avec la même caméra. Le plan est souvent l'unité temporelle la plus petite pour une séquence vidéo si l'on ne prend pas en compte l'image pour laquelle la notion de temps a disparu.

Chaque plan est séparé du précédent et du suivant par une transition. On distingue deux types de transitions : les transitions brusques et les transitions progressives (cf. figure 4). Lors d'une transition brusque (appelée *cut*), la dernière image du premier plan est directement suivie par la première image du second plan. Dans le cas où les deux plans sont connectés en utilisant un effet particulier, on parle de transition progressive. Différents types de transitions peuvent être utilisés. Les plus connus sont le fondu et le volet. Au cours d'un fondu, le niveau de chaque pixel des images intermédiaires est calculé en fonction des

niveaux des pixels de la dernière image du premier plan et de la première image du second plan. L'influence varie au cours de la transition de 0 à 1 pour la première image du second plan et de 1 à 0 pour la dernière image du premier plan. Lors d'un volet, chaque pixel des images intermédiaires a un niveau égal à celui du pixel de mêmes coordonnées spatiales soit dans la dernière image du premier plan soit dans la première image du second plan. La variation des proportions de pixels de chaque image est monotone. Les logiciels actuels permettent des effets de plus en plus sophistiqués mais qui conservent néanmoins ces principes de base.

Nous allons maintenant voir brièvement les différentes familles de méthodes existantes pour détecter ces différents changements.

4.2. Description de quelques solutions classiques

La plupart des méthodes proposées pour résoudre le problème de la détection des changements de plan fonctionnent en deux étapes : le calcul d'une mesure de dissimilarité entre deux trames successives d'une séquence vidéo, puis la comparaison de la valeur obtenue avec un seuil, afin de déterminer ou non la présence d'un changement de plan. Suivant ce principe, la détection d'un changement de plan est effective si la condition suivante est vérifiée :

$$d(I_t, I_{t-1}) > S \quad (23)$$

Nous rappelons que I_t représente l'image de la séquence vidéo obtenue à l'instant t , d une distance, et S un seuil.

Le problème de la détection des changements de plan a été étudié de manière approfondie et de nombreuses méthodes ont été proposées. Il n'est donc pas possible de dresser ici un panorama complet de l'existant, et le lecteur est invité à consulter [11] pour un état de l'art du domaine. Nous pouvons néanmoins établir une classification hiérarchique de ces méthodes, en se basant tout d'abord sur le type de séquences vidéo (compressées ou non) et ensuite sur le type d'informations utilisées pour réaliser la segmentation. Dans le cas des séquences vidéo non-compressées, on dispose des valeurs (en niveaux de gris ou en couleur) des pixels composant les différentes images de la séquence vidéo. Dans ce contexte, il est possible de se baser directement sur les valeurs des pixels et de mesurer la somme des différences absolues terme à terme pour des images en niveaux de gris ou en couleur, ou alors de compter le nombre de couples (pixels de coordonnées spatiales identiques mais de coordonnées temporelles voisines) ayant des intensités éloignées. Ce type de méthode, quoique simple, s'est avéré peu robuste au bruit et aux mouvements présents dans la séquence. Certains auteurs ont donc proposé d'utiliser des statistiques plus globales comme les histogrammes. Il est alors possible de calculer une différence terme à terme en considérant les classes de l'histogramme afin de comparer deux images successives. Cependant ces méthodes souffrent de leur caractère purement global. Par conséquent, une analyse à un niveau intermédiaire (entre local et global) a été étudiée. Les traitements s'effectuent au niveau des blocs, et la comparaison de deux images s'effectue en mesurant les différences pour chaque paire de blocs *via* un calcul de la moyenne et de l'écart-type des intensités pour chaque bloc. Une fois chaque paire de blocs étiquetée selon sa similarité, la détection dépend du nombre de paires de blocs différents. De



Figure 4. Exemple de transitions (de haut en bas) : cut, fondu, et volet.

par la nature temporelle de la séquence d'images, il est également possible de se baser sur l'information de mouvement, mais celle-ci nécessite généralement un coût important pour être obtenue. Finalement des auteurs ont proposé de combiner différentes approches, permettant ainsi de combiner les avantages de chacune d'entre elles. Cependant le temps de calcul s'accroît considérablement avec la complexité des opérations à effectuer. Nous avons mentionné ici les principales familles de solutions permettant la détection des changements de plan, dont un panorama est donné en [11]. Nous allons maintenant décrire la méthode proposée, qui peut s'appliquer à des données compressées ou non grâce à un prétraitement approprié des données. En outre, nous verrons qu'elle peut s'adapter au type de séquences vidéo analysées et qu'elle s'exécute en temps-réel.

4.3. Prétraitement des données

Afin de garantir un temps de calcul relativement faible, et de considérer indifféremment des données compressées ou non, nous proposons d'introduire un prétraitement des données. Celui-ci consistera à diminuer la résolution spatiale des images à analyser. Ainsi, le nombre de pixels à traiter sera moins élevé, diminuant alors le nombre de calculs nécessaires. Nous ramenons à des images de superficie 64 fois inférieure à celle des images originales. La résolution atteinte permet encore visuellement de détecter les changements de plan.

Chaque pixel est caractérisé par une valeur pour chaque composante couleur de la représentation utilisée. Le but de l'étape décrite ici est de représenter chaque image par un nombre de pixels inférieur à celui d'origine. Nous proposons donc de créer à partir de chaque image originale I (contenant $X \times Y$ pixels) une nouvelle image I' , composée de $X' \times Y'$ pixels. On pose $X' = X/8$ et $Y' = Y/8$. Pour chaque bloc 8×8 (64 pixels) et pour chaque composante couleur, la moyenne du bloc est calculée. Dans le cas de données compressées (JPEG ou MPEG), la moyenne est assimilée au coefficient de basse-fréquence DC. Les valeurs obtenues permettent de caractériser les nouveaux pixels de l'image à faible résolution.

Une fois les données de chaque image normalisées et réduites, les images successives vont être comparées au moyen d'une mesure de distance. Celle-ci va maintenant être introduite.

4.4. Mesures de distance

De manière à accroître la robustesse aux changements d'illumination et aussi à réduire le temps de calcul nous avons choisi de limiter la représentation des pixels à un espace de dimension 2 composé de la teinte et de la saturation. Pour toutes les scènes d'extérieur, qui sont fréquentes dans les séquences vidéo, on peut noter une amélioration par rapport aux résultats obtenus dans l'espace RVB, comme nous le verrons dans la section 5.

Comme [2, 6], nous mesurons la différence entre deux images dans le sous-espace TS. Nous proposons ici la définition d'une distance d :

$$d(I_{t_1}, I_{t_2}) = \sum_{x=1}^X \sum_{y=1}^Y I_{t_1}(x, y) \ominus I_{t_2}(x, y) \tag{24}$$

avec \ominus un opérateur algébrique utilisé pour comparer deux pixels et défini par :

$$I_{t_1}(x, y) \ominus I_{t_2}(x, y) = \alpha_{T,S}(I_{t_1}(x, y, T) - I_{t_2}(x, y, T)) \bmod 2\pi + (1 - \alpha_{T,S})|I_{t_1}(x, y, S) - I_{t_2}(x, y, S)| \tag{25}$$

où $\alpha_{T,S}$ est un coefficient permettant de donner plus ou moins d'influence aux composantes T et S. En effet, dans le cas de pixels achromatiques, il est important de ne pas donner beaucoup d'importance à la composante T qui n'est alors pas fiable. Ce coefficient est défini par :

$$\alpha_{T,S} = c \cdot \chi_{S > S_S} \text{ avec } \chi_P = \begin{cases} 1 & \text{si } P \text{ est vérifiée} \\ 0 & \text{sinon} \end{cases} \tag{26}$$

avec c une constante et S_S un seuil comparé aux valeurs de saturation. Ce coefficient, propre à chaque pixel, est donc indépendant du type d'image considéré.

La mesure de distance d , quoique relativement simple, permet d'estimer correctement la différence entre deux images en assurant une invariance à l'illumination. L'utilisation d'une mesure de distance plus complexe pourrait apporter une qualité d'information supérieure mais engendrerait également un surcoût en terme de temps de calcul. Cependant, l'utilisation directe de cette mesure de dissimilarité entre deux images successives nécessite la comparaison à un seuil S (cf. équation (23)). Le seuil utilisé doit souvent être fixé de manière empirique, et dépend du domaine vidéo étudié (sport, bulletin d'informations, etc.) ou du type de plans présents dans la séquence. Ainsi, un plan éloigné, où les objets en mouvement sont petits, sera caractérisé par une valeur d relativement faible tandis qu'un plan proche ou serré, où les objets en mouvement occupent une portion importante de l'image, sera caractérisé par une valeur d plus élevée. Le seuil S devra donc être ajusté en conséquence afin d'éviter les fausses détections ou l'absence de détection. Comme certaines séquences vidéo, notamment les retransmissions télévisées d'événements sportifs, contiennent des plans éloignés et des plans proches, il est nécessaire d'utiliser une méthode plus générale qui puisse s'adapter à ces différents types de plans. Nous proposons simultanément l'utilisation d'un seuil adaptatif et d'une mesure différentielle. En introduisant un seuil adaptatif, noté S_d , qui évolue au cours du temps, la valeur du seuil est mise à jour pour chaque nouvelle image de la séquence (permettant ainsi de limiter les faux positifs et les faux négatifs), suivant la formule :

$$S_d(t) = \alpha_{S_d} S_d(t-1) + (1 - \alpha_{S_d}) d(I_t, I_{t-1}) \quad (27)$$

où $S_d(t)$ représente la valeur du seuil S_d à l'instant t . De cette façon, il s'adapte automatiquement avec une certaine inertie (représentée par le coefficient α_{S_d}) au contenu de la vidéo étudiée, sa valeur étant modifiée en fonction des valeurs de $d(I_t, I_{t-1})$. Le paramètre α_{S_d} a une influence directe sur les mesures de précision et de rappel, tandis que le choix initial de $S_d(0)$ a très peu d'incidence sur les résultats.

L'utilisation directe d'une mesure d entre deux images successives est très sensible au bruit et au mouvement présent dans la séquence étudiée. L'introduction d'un seuil adaptatif permet de limiter cette sensibilité dans une certaine mesure, mais évidemment pas dans sa totalité. Nous proposons donc de considérer une mesure relative et non une mesure absolue. Cette mesure relative, notée d' , permet d'accroître la robustesse au bruit et aux mouvements importants présents dans la séquence, et est définie par :

$$d'(I_t) = |d(I_t, I_{t-1}) - d(I_{t-1}, I_{t-2})| \quad (28)$$

Nous avons ainsi une estimation de la dérivée seconde du signal vidéo. Contrairement à la mesure d , la mesure d' est définie de manière relative et son ordre de grandeur dépend donc moins du type de plan ou de vidéo étudié. Afin de détecter un changement de plan, cette mesure peut donc être comparée à un seuil $S_{d'}$ fixé empiriquement au début de la séquence. La valeur de $S_{d'}$ pourra évoluer en fonction du type de vidéo ou de plan analysé. Cependant, nous n'avons pas observé expérimentalement le besoin d'adapter la valeur $S_{d'}$ aux différentes données (de différents domaines : sport, films, etc.) sur lesquelles la méthode a été appliquée, ce qui nous permet d'affirmer une certaine robustesse de notre approche.

Nous avons présenté ici la mesure de dissimilarité utilisée et justifié l'intérêt d'utiliser la variation d'une mesure de distance calculée entre deux images successives plutôt que la mesure de distance elle-même. Nous présentons maintenant l'algorithme de détection des transitions brusques ou progressives.

4.5. Méthode de segmentation

Comme il a été précisé précédemment, un changement de plan peut être brusque ou progressif. En considérant *une transition progressive comme une transition brusque dont les effets sont étalés sur plusieurs images*, nous proposons ici une approche permettant de détecter les transitions brusques ou progressives de manière relativement similaire.

Pour détecter un changement brusque, nous comparons directement la valeur d' avec un seuil $S_{d'}$. En effet, si la valeur de d' est élevée, c'est-à-dire si la différence absolue entre $d(I_t, I_{t-1})$ et $d(I_{t-1}, I_{t-2})$ est significative, alors l'évolution du contenu de la séquence entre les images I_{t-1} et I_t n'est pas cohérente avec

celle observée entre les images I_{t-2} et I_{t-1} . Un changement brusque existe donc à l'instant $t-1$.

Si aucun changement brusque n'a été détecté, il est encore possible de se trouver en présence d'une transition progressive. La valeur d' ne peut être utilisée directement dans le cas d'une transition progressive puisqu'elle ne reflète l'évolution de la mesure de distance d qu'à un instant donné. Il est donc nécessaire de cumuler les valeurs d' obtenues pour toutes les trames composant une transition progressive afin d'obtenir une mesure qui sera du même ordre de grandeur que la valeur du seuil considéré dans le cas d'une transition brusque.

La détection des transitions progressives s'effectue donc en deux étapes successives. La première étape consiste en la détection des trames susceptibles d'être les frontières d'une transition progressive. Pour détecter celles-ci, nous analysons l'évolution de la mesure de distance d et nous comparons à chaque instant t la valeur $d(I_t, I_{t-1})$ avec le seuil adaptatif $S_d(t)$ défini par l'équation (27). L'utilisation de ce seuil adaptatif nous permet de gérer tout type de situation (plan proche ou éloigné, mouvement important ou pas, etc.). Si la condition définie par $d(I_t, I_{t-1}) > S_d(t_1)$ est vérifiée, alors il est possible qu'une transition soit présente dans la séquence à partir de l'instant t_1 . L'instant t_2 de fin de cette transition correspondrait à la première trame vérifiant la condition $d(I_{t_2}, I_{t_2-1}) < S_d(t_2)$ avec $t_2 > t_1$. Une fois les frontières t_1 et t_2 d'une possible transition déterminées, il est nécessaire d'analyser les trames t de cet intervalle de temps. Pour cela, nous calculons la valeur cumulée de d' sur l'ensemble des trames $[t_1, t_2]$. Elle est notée $d'_{\text{cumul}}(t_1, t_2)$:

$$d'_{\text{cumul}}(t_1, t_2) = \sum_{t=t_1}^{t_2} d'(I_t) \quad (29)$$

La comparaison de $d'_{\text{cumul}}(t_1, t_2)$ avec le seuil $S_{d'}$ permet alors de valider ou non la présence d'un changement progressif entre les trames I_{t_1} et I_{t_2} . Cette approche par dérivation / intégration nous permet de nous affranchir des valeurs initiales.

La méthode de segmentation temporelle décrite ici, ainsi que la méthode de segmentation spatiale introduite précédemment, ont toutes deux été testées dans le contexte de l'analyse de séquences vidéo de football afin de vérifier leur efficacité. Les résultats obtenus vont maintenant être présentés. Ils dépendent fortement de l'espace de représentation que nous avons considéré.

5. Résultats

Afin d'illustrer l'intérêt de l'espace TSL dans les problèmes de segmentation, nous avons décrit précédemment deux méthodes basées sur cet espace couleur et permettant respectivement la segmentation spatiale et la segmentation temporelle de données image. Nous présentons ici quelques résultats obtenus avec ces méthodes, nous permettant ainsi d'illustrer l'intérêt de l'espace

TSL et les capacités des méthodes proposées. Le contexte considéré est celui de l'analyse temps-réel de séquences vidéo de matchs de football. Les images analysées représentent des scènes d'extérieur où l'illumination n'est pas constante. Les mesures de temps de calcul ont été obtenues en considérant une architecture PC Pentium 4 cadencé à 1.7 GHz.

5.1. Segmentation spatiale

La méthode de segmentation spatiale a pour but de séparer les objets et le fond. Dans le contexte proposé, l'objectif est d'identifier les joueurs par rapport au terrain de football. La figure 5 montre les résultats obtenus sur deux images différentes où les tailles des objets varient considérablement. Les objets sont correctement détectés, indépendamment de l'aire qu'ils occupent dans l'image. À titre de comparaison, les résultats obtenus après expérimentation d'une approche de seuillage automatique d'histogramme [13] par la méthode de Ridler sont également donnés : on peut constater que les objets représentés par des régions de plusieurs couleurs ne sont pas détectés correctement. Cette comparaison nous permet de souligner la qualité de notre approche même si les frontières des régions obtenues sont imprécises (effet de bloc). Pour notre approche, les temps de calcul observés dépendent de la taille de l'image : de 30 millisecondes pour une image de taille 192×128 pixels à 350 millisecondes pour une image de taille 704×576 pixels.

La segmentation a été obtenue en utilisant les paramètres présentés dans le tableau 1. La structure de la pyramide est régulière, nous avons donc p^2 et K_0 égaux à 4. Les seuils S_1 et S_2 ont été obtenus empiriquement mais le processus de segmentation est robuste à leur réglage, comme nous pourrions le constater par la suite (figure 9).

Tableau 1. Paramètres utilisés lors de la séparation du fond et des objets par approche multirésolution.

Paramètre	Description	Valeur
r_{max}	Nombre de couches de la pyramide	7
r_{final}	Résolution du résultat final	5
p^2	Nombre de pixels utilisés à la résolution r pour générer un pixel à la résolution $r + 1$	4
K_0	Constante influant sur le nombre de zones	4
S_1	Seuil utilisé lors de l'appariement	$\pi/18$
S_2	Seuil utilisé lors du test d'homogénéité	$\pi/4$

Le processus de segmentation est itératif. La figure 6 illustre la diminution progressive de la résolution pour obtenir le modèle initial de l'arrière-plan, tandis que la figure 7 présente le résultat de la segmentation aux différentes résolutions. On peut observer que le choix de la résolution finale r_{final} influe directement sur la précision du résultat. Cependant, même en consi-

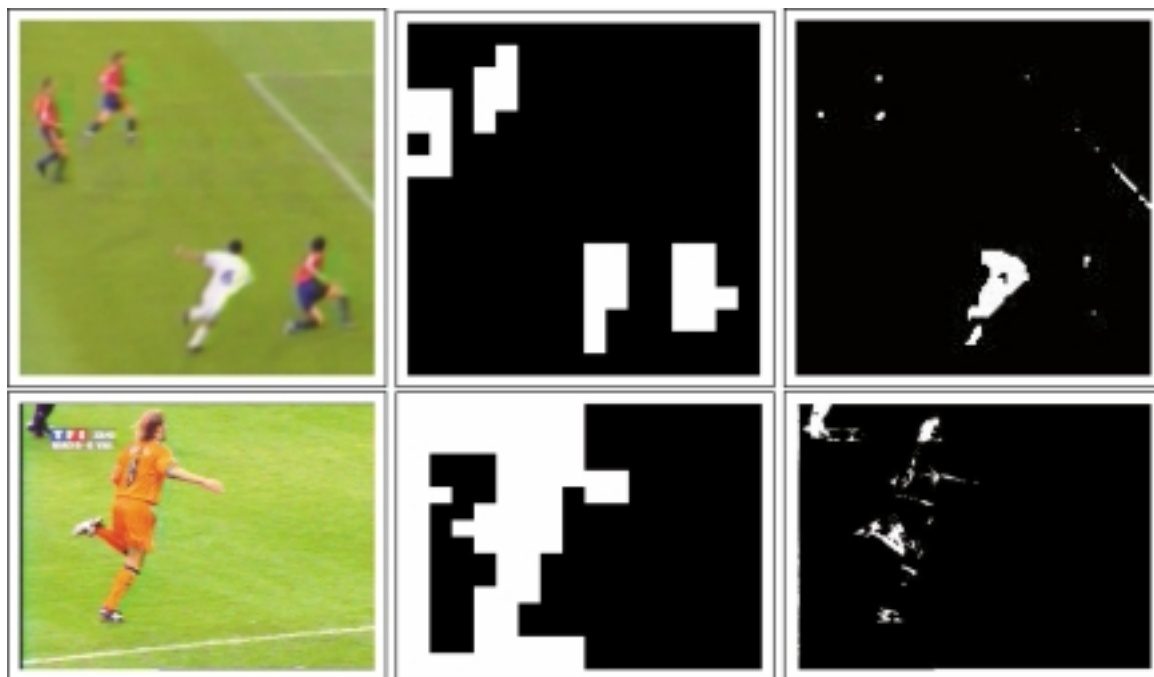


Figure 5. Résultats obtenus en considérant des objets de différentes tailles (à gauche), avec l'approche proposée (au centre) et une technique de seuillage automatique d'histogramme [13] par la méthode de Ridler (à droite). Dans les deux cas, les zones étiquetées comme « objet » sont représentées en blanc.

dérant une résolution finale similaire à la résolution originale (i.e. $r_{\text{final}} = 0$), les contours des objets détectés resteront grossiers et parallèles aux côtés de l'image. La séparation des objets et du fond n'étant généralement qu'une étape intermédiaire (permettant l'initialisation d'algorithmes de suivi d'objets), la précision obtenue est néanmoins suffisante.

La figure 8 illustre les résultats obtenus en considérant la teinte mais aussi l'intensité et les composantes rouge, vert et bleu de l'espace RVB. Pour ces dernières composantes, les valeurs des paramètres S_1 et S_2 ont été adaptées à l'intervalle de définition

(de 2π à 255). L'utilisation de la teinte fournit un résultat de meilleure qualité. Aucune des composantes R, V ou B ne contient l'information suffisante pour atteindre des résultats aussi précis qu'avec la teinte (l'information pertinente étant dispersée dans les trois canaux de base). De plus, une combinaison classique de ces trois composantes sous la forme d'une intensité n'apporte pas de meilleur résultat. Le choix d'une représentation couleur adaptée est donc particulièrement important pour obtenir un résultat satisfaisant.



Figure 6. Représentation d'une image à différentes résolutions, de la résolution originale $r = 0$ (à gauche) à $r = r_{\text{max}} = 5$ (à droite).

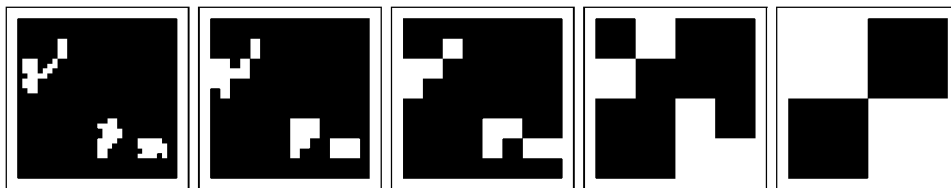


Figure 7. Résultat obtenu aux différentes résolutions, de $r_{\text{final}} = r_{\text{max}} - 1 = 4$ (à droite) à $r_{\text{final}} = 0$ (à gauche).

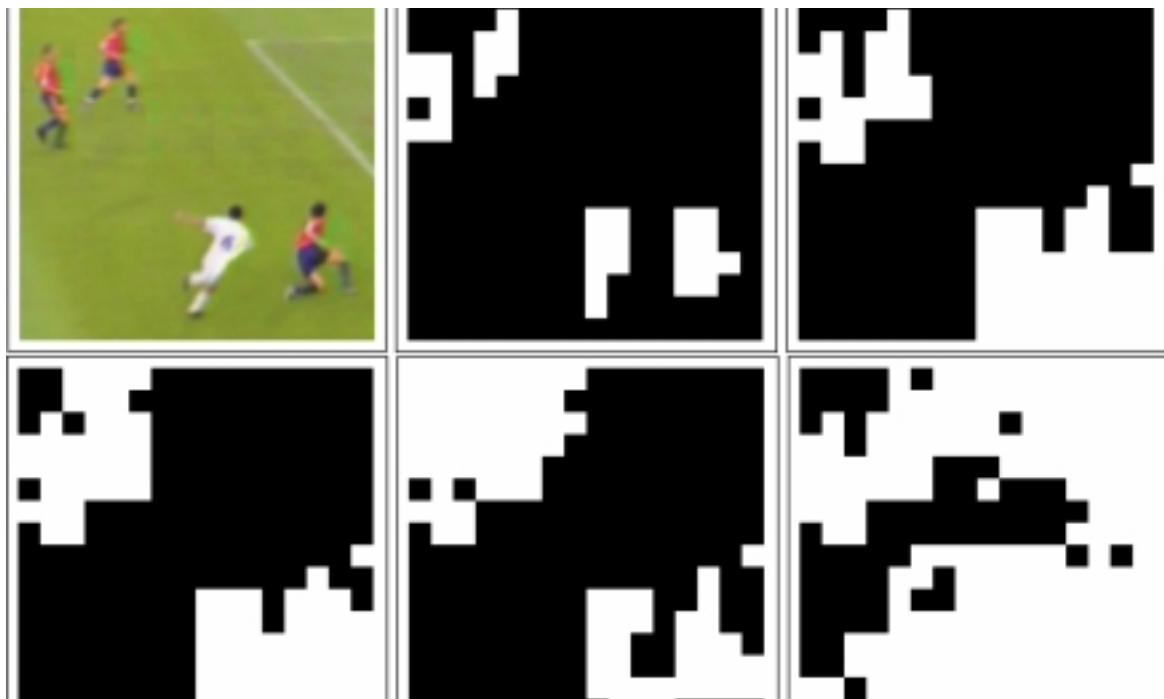


Figure 8. Résultats obtenus pour une image de taille 128×128 pixels (en haut à gauche) en considérant la teinte (en haut au milieu), l'intensité (en haut à droite) ou les composantes de l'espace RVB (en bas): rouge, vert, et bleu (de gauche à droite).

Une seconde comparaison est donnée dans la figure 9. Ici différents jeux de paramètres ont été testés afin d'évaluer la robustesse de la composante couleur utilisée pour le réglage des paramètres. Là encore, la teinte fournit les résultats les plus intéressants et les plus robustes, comparativement à l'intensité ou aux composantes rouge, vert et bleu. Les paramètres ont varié du simple au quadruple, c'est-à-dire (S_1, S_2) , $(2S_1, 2S_2)$, $(3S_1, 3S_2)$ et $(4S_1, 4S_2)$.

Les principales limites de la méthode proposée ont été identifiées *a priori* dans la section 3. D'une part, les pixels ne doivent pas être achromatiques (puisque seule la teinte est utilisée dans le processus de segmentation), et d'autre part l'arrière-plan doit être relativement uniforme (puisque'il est modélisé ensuite par une moyenne). L'influence de la contrainte liée à la chromaticité des images, mesurée par le niveau de saturation des pixels, a été évaluée sur les images de la figure 10. Nous avons observé

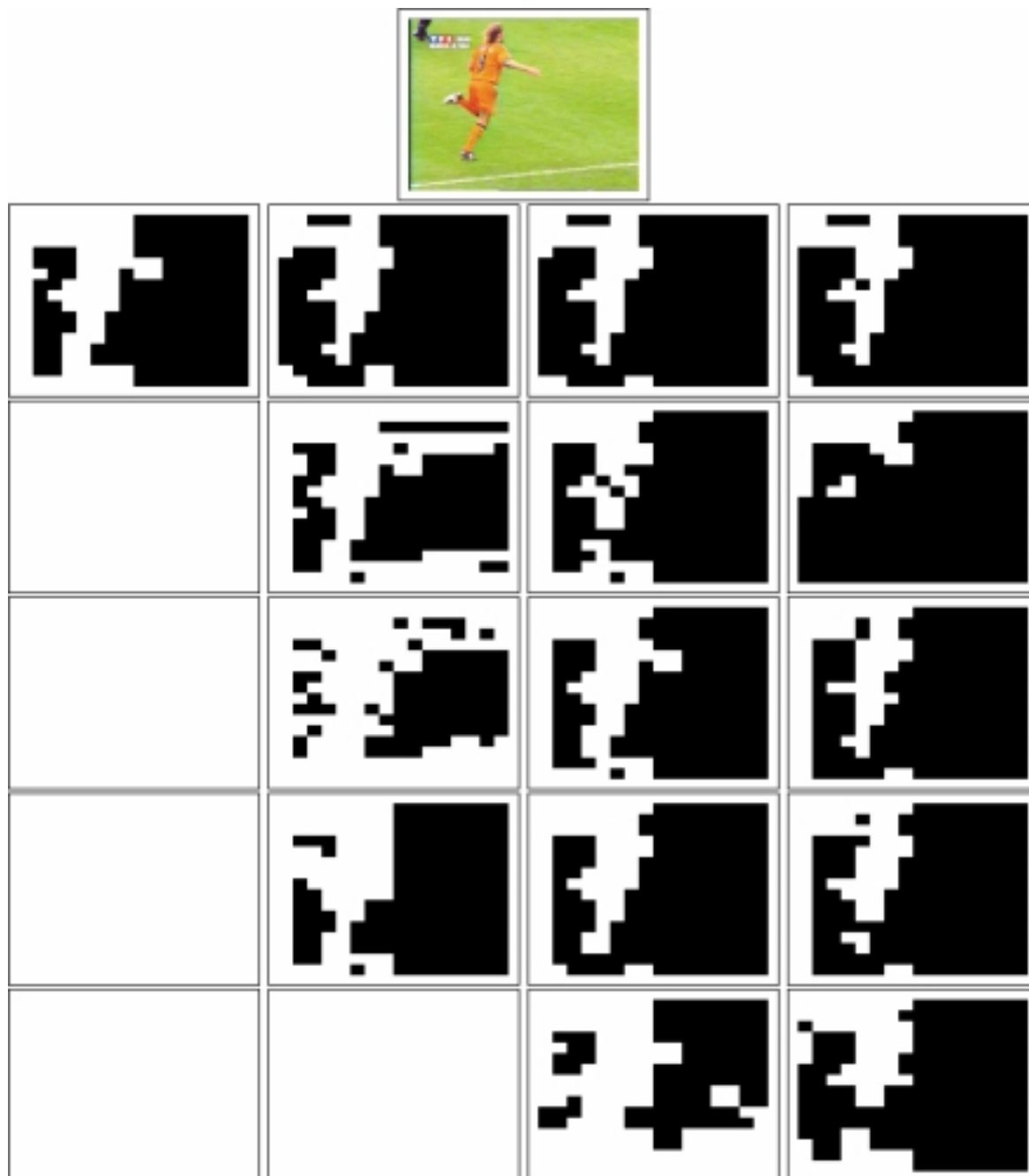


Figure 9. Résultats obtenus pour une image de taille 384×288 (première ligne) en considérant la teinte, l'intensité, et les composantes de RVB (de haut en bas : teinte, intensité, rouge, vert, bleu) avec des jeux de paramètres variables (de gauche à droite : facteur 1, facteur 2, facteur 3, facteur 4).

qu'une baisse de 25 ou 50 % du niveau de saturation n'avait pas d'influence sur les résultats obtenus, tandis qu'une baisse de saturation de 75 % nécessite un nouveau réglage de paramètre (seuil $S_1 = \pi/6$, soit 3 fois supérieur au seuil initial). Lorsque la saturation est nulle (image de droite de la figure 10), la méthode ne donne évidemment pas de résultat pertinent. Cependant, très peu d'images naturelles représentant des scènes d'extérieur et fournies par des capteurs couleur sont achromatiques.

Nous allons maintenant décrire les résultats obtenus par la méthode de segmentation temporelle.

5.2. Segmentation temporelle

Nous avons évalué notre méthode en terme de qualité et d'efficacité. Pour mesurer la qualité d'une méthode de détection de changements de plan, il est possible d'utiliser des mesures (ou taux) de rappel T_r et de précision T_p . Nous rappelons que ces mesures sont définies par :

$$T_r = \frac{N_d}{N_d + N_m} \quad (30)$$

$$T_p = \frac{N_d}{N_d + N_f} \quad (31)$$

où N_d , N_m , et N_f représentent respectivement le nombre de détections correctes, de détections manquées (ou faux négatifs), et de fausses détections (ou faux positifs).

La méthode proposée ici a été testée sur des séquences vidéo de différentes origines. Afin de souligner ses capacités, nous insistons ici sur les résultats obtenus à partir de retransmissions de matchs de football et considérons ce contexte comme plus difficile que celui des journaux télévisés par exemple. En effet, les

séquences analysées sont caractérisées par des plans proches ou éloignés, un grand nombre d'effets différents utilisés, une variation importante des mouvements de la caméra et des objets présents dans la scène, des changements d'illumination fréquents liés au système d'éclairage, et enfin une certaine homogénéité des images de la séquence vidéo (même si elles appartiennent à différents plans). La première ligne du tableau 2 présente les taux de détection dans le cas de journaux télévisés et de séquences de football : les mesures obtenues illustrent bien la difficulté à traiter correctement les séquences de football.

Nous avons également comparé dans ce même tableau les résultats obtenus par notre approche avec ceux donnés par deux techniques classiques [11] : la première, basée sur une différence pixel-à-pixel, montre l'incompatibilité de ce type d'approche avec des scènes acquises avec une caméra en mouvement, tandis que la seconde, basée sur une différence d'histogrammes, illustre la difficulté à traiter des scènes à fond globalement uniforme. Les méthodes classiques proposées dans la littérature, même si elles fournissent des résultats intéressants sur des corpus usuels (de type journal télévisé), montrent bien ici leur incompatibilité avec des données plus complexes telles que des retransmissions de football. Notre méthode, quant à elle, fournit des résultats particulièrement intéressants, et s'affranchit très fortement du besoin d'adaptation des seuils à chaque nouvelle séquence traitée.

Les résultats présentés ici ont été obtenus après une étape manuelle de réglage des paramètres optimaux en considérant 15 % du corpus de test. Le corpus contient une centaine de changements de plan (un quart de transitions brusques et trois quarts de transitions progressives) pour environ 12000 trames, réparties à parts égales entre journal télévisé (extrait du corpus INA) et football. Enfin, notre approche a été paramétrée avec une pondération équivalente de la teinte et de la saturation, une saturation significative fixée à 25 %, et une inertie importante

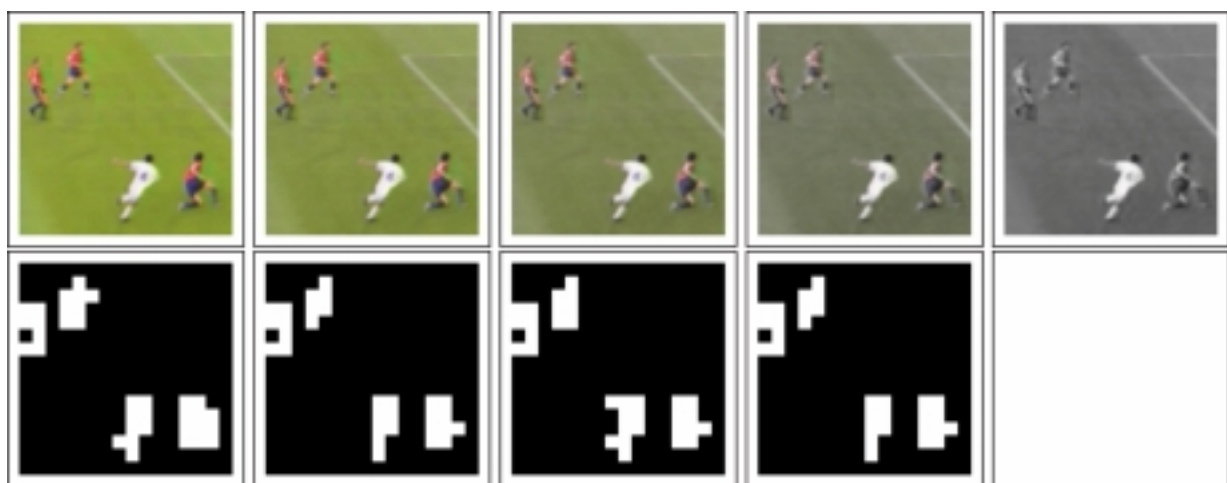


Figure 10. Image analysée (en haut) et résultat obtenu (en bas), en considérant une saturation décroissante (de gauche à droite) : 0 % (niveau original), 25 %, 50 %, 75 %, et 100 %.

du seuil adaptatif. Ces paramètres sont décrits dans le tableau 3. Les mesures S_S et c liées respectivement à la fiabilité d'une teinte et à la pondération entre teinte et saturation ont été définies en se basant sur les travaux de Carron [2].

Tableau 2. Qualité de la détection des changements de plan dans l'espace TSL.

Méthode utilisée	Journal		Football	
	Rappel	Précision	Rappel	Précision
Approche proposée	84 %	98 %	72 %	88 %
Approche basée pixel	90 %	70 %	33 %	33 %
Approche basée histogramme	39 %	100 %	12 %	24 %

Tableau 3. Paramètres utilisés lors de la détection des changements de plan.

Paramètre	Description	Valeur
c	Constante utilisée dans le calcul de $\alpha_{T,S}$	0,5
S_S	Seuil utilisé pour déterminer la fiabilité d'une teinte	0,25
α_{S_d}	Inertie du seuil adaptatif S_d	0,75

Après l'étape de réduction spatiale d'un facteur 8×8 , les images à traiter ne contiennent plus que 20×15 pixels au lieu des 160×120 originels, comme le montre la figure 11. Les séquences vidéo contiennent différents effets (cut, volet, fondu, mais aussi certains effets combinant volet et fondu). Elles incluent aussi du bruit dû au mouvement global de la caméra,

aux objets en mouvement, ainsi qu'aux effets d'illumination. Pour la taille d'image considérée (160×120 pixels), le temps de calcul nécessaire est égal à 4 millisecondes par image en considérant la même architecture matérielle que précédemment. Les figures 12 à 14 montrent l'évolution des mesures d , d' , et d'_{cumul} pour des séquences vidéo contenant différents types de transition, en considérant d'une part l'espace TSL et d'autre part l'espace RVB. Dans le premier exemple (figure 12, cas de deux cuts), l'utilisation de l'espace RVB entraîne de nombreuses fausses détections. Dans le second exemple (figure 13, cas d'un volet), les résultats obtenus avec les deux espaces sont comparables. Enfin, dans le dernier exemple (figure 14, cas d'un fondu), la transition n'est pas détectée dans l'espace RVB tandis que de fausses détections sont identifiées. On observe donc sur les 3 exemples représentatifs que le contraste des valeurs au niveau des extremums locaux est beaucoup plus marqué dans l'espace TSL que dans l'espace RVB. Ce choix d'espace assure donc à la méthode proposée une plus grande robustesse et confirme notre hypothèse théorique de départ.

La principale limite de l'approche proposée ici est sa sensibilité au mouvement présent dans la séquence. Ce mouvement apparent peut être provoqué par une accélération brusque de la caméra ou par le mouvement d'un objet occupant quasiment toute l'image dans un plan rapproché. Il est donc nécessaire de trouver un compromis entre fausses détections liées au mouvement et détections manquées des effets de type fondu. Néanmoins il semble que nous ayons repoussé les limites acceptables en ayant recours à une différence du second ordre ainsi qu'à un seuil adaptatif. De plus, nous pouvons remarquer que les mesures de rappel et de précision sont supérieures à 90 % si seules les transitions brusques sont considérées (alors que pour ces mêmes transitions, les mesures obtenues par les approches basées pixel ou histogramme varient entre 10 et 25 %).

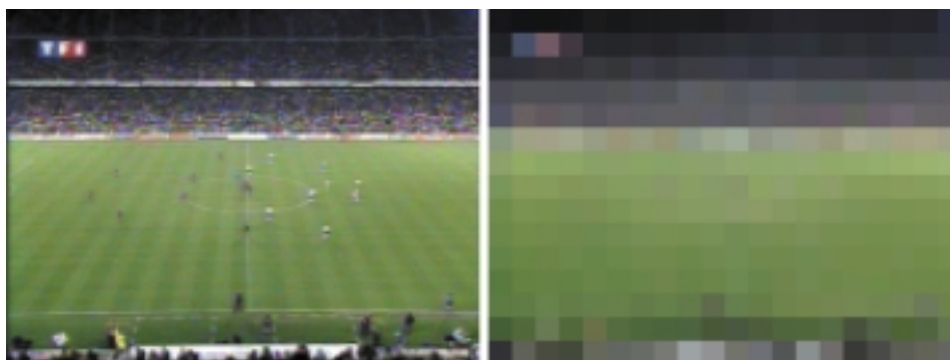


Figure 11. Réduction de la résolution spatiale d'une image par un facteur 64 : image originale (à gauche) et image réduite (à droite).

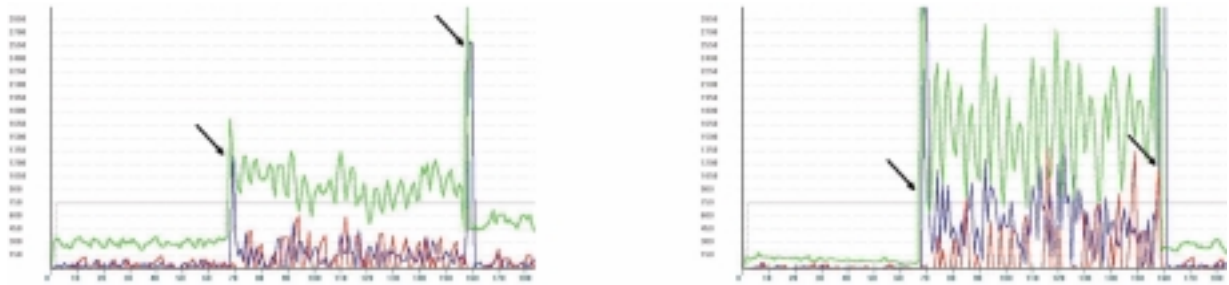


Figure 12. Évolution temporelle des mesures d (en vert), d' (en rouge), et d'_{cumul} (en bleu) pour une séquence contenant des transitions brusques indiquées par les flèches, en se basant sur l'espace TSL (à gauche) ou RVB (à droite).

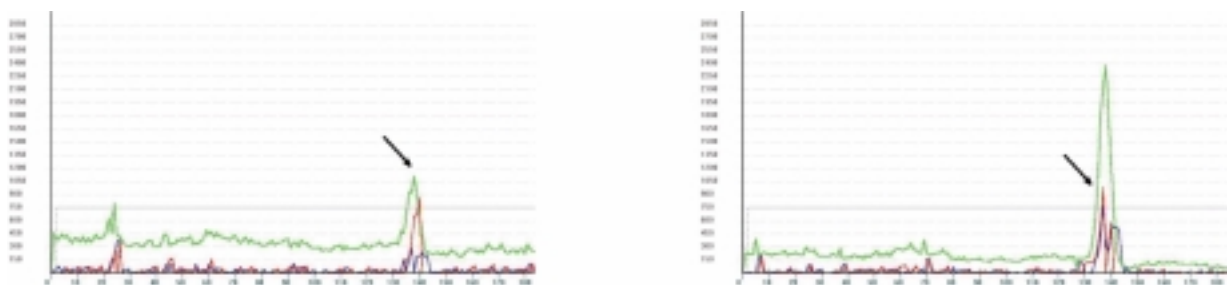


Figure 13. Évolution temporelle des mesures d (en vert), d' (en rouge), et d'_{cumul} (en bleu) pour une séquence contenant une transition progressive de type volet indiquée par la flèche, en se basant sur l'espace TSL (à gauche) ou RVB (à droite).

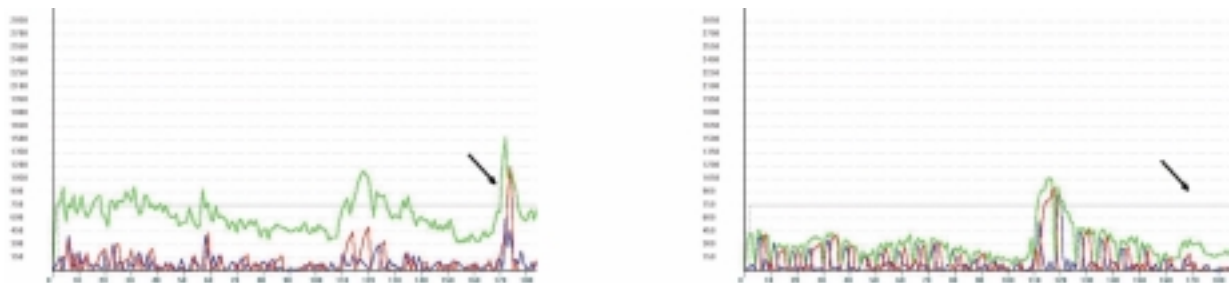


Figure 14. Évolution temporelle des mesures d (en vert), d' (en rouge), et d'_{cumul} (en bleu) pour une séquence contenant une transition progressive de type fendu indiquée par la flèche, en se basant sur l'espace TSL (à gauche) ou RVB (à droite).

6. Conclusion

La couleur est de plus en plus souvent utilisée en traitement d'images. Cependant, l'espace de représentation généralement considéré est l'espace Rouge Vert Bleu prévu pour l'affichage des images à l'écran. Nous avons montré ici comment un autre espace de représentation de la couleur, l'espace Teinte Saturation Luminance, pouvait apporter une amélioration en traitement d'images. Pour cela nous avons considéré le problè-

me de la segmentation, successivement dans l'espace et dans le temps, et y avons apporté des solutions originales basées sur l'espace TSL.

Dans cet article, nous avons tout d'abord identifié les caractéristiques et spécificités de l'espace TSL et donné quelques modes de calcul relatifs à la teinte. Nous avons ensuite introduit une méthode originale de segmentation spatiale qui ne nécessite qu'une seule image grâce à un cadre d'analyse multirésolution. En complément, nous avons proposé une méthode de segmentation temporelle s'adaptant au contenu, notamment par

l'usage d'une mesure de distance intertrames différentielle. Notre contribution a donc porté sur ces trois aspects : caractéristiques et calculs dans l'espace TSL, segmentation spatiale ou séparation des objets et du fond, segmentation temporelle ou détection des changements de plan.

Les perspectives des travaux présentés ici sont de deux ordres. Tout d'abord il nous paraît important d'approfondir les résultats présentés, en continuant d'identifier les caractéristiques de l'espace TSL et les méthodes de calcul appropriées à cet espace. De plus, nous souhaitons atténuer les limites des méthodes proposées. Dans le cas de la segmentation spatiale, nous pensons que les pixels achromatiques pourraient être pris en compte en modifiant la modélisation pour y intégrer les informations de teinte et de saturation (sous une forme pondérée, comme dans le cas de la segmentation temporelle). La prise en compte de scènes caractérisées par un arrière-plan plus complexe (par exemple non plus uniforme mais composé de parties uniformes) passe par l'intégration de plusieurs valeurs (moyenne des teintes de chaque zone) dans le modèle et par la propagation de ce modèle complexe dans les différentes résolutions. En ce qui concerne la segmentation temporelle, une meilleure prise en compte des artefacts liés au mouvement doit être effectuée. Elle pourrait résulter d'une identification précise des effets du mouvement sur la mesure de distance proposée afin de limiter ces effets.

Outre les possibilités de futures recherches décrites précédemment, nous pensons qu'il serait intéressant également de montrer l'intérêt de l'espace couleur TSL au travers d'autres applications, comme la détection de contours ou le codage par exemple. De plus, une étude comparant les avantages et inconvénients des différents modèles TSL proposés dans la littérature nous semble tout à fait appropriée et permettrait de choisir l'espace couleur approprié pour un problème donné en analyse et traitement des images.

Références

- [1] R. BEUN, Image based camera motion stabilization using static features in the distant scene. In *International Workshop on Advanced Concepts for Intelligent Vision Systems*, pages 276-283, Gand, Belgique, Septembre 2002.
- [2] T. CARRON, *Segmentations d'images couleur dans la base Teinte-Luminance-Saturation : approche numérique et symbolique*. Thèse de doctorat, Université de Savoie, Décembre 1995.
- [3] H.D. CHANG, X.H. JIANG, Y. SUN et J. WANG, Color image segmentation: Advances and prospects. *Pattern Recognition*, 34(12):2259-2281, Décembre 2001.
- [4] M. CIESIELSKA, Testing uniformity for circular data. Mémoire de maîtrise, West Virginia University, Mai 2001.
- [5] B. ELLIGER, Analysis of motion compensated filters concerning motion correctness and accuracy. *Signal Processing: Image Communication*, 14(9):697-720, Juillet 1999.
- [6] C. GARBAY, *Modélisation de la couleur dans le cadre de l'analyse d'images et de son application à la cytologie automatique*. Thèse de doctorat, Institut National Polytechnique de Grenoble, 1979.
- [7] R.C. GONZALEZ et R.E. WOODS, *Digital Image Processing*. Addison-Wesley, 1992.
- [8] P. GOUTON, P. BONTON et P. LAMBERT editors, *Traitement du Signal, Numéro Spécial sur l'Imagerie Numérique Couleur*, 21(5-6), 2004.
- [9] B.G. HASKELL *et al.* Image and video coding – emerging standards and beyond. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(7):814-837, Novembre 1998.
- [10] J.M. JOLION et A. ROSENFELD, *A Pyramid Framework for Early Vision*. Kluwer Academic Publishers, Décembre 1993.
- [11] S. LEFÈVRE, J. HOLLER et N. VINCENT, A study of real-time segmentation of uncompressed video sequences for content-based search and retrieval. *Real-Time Imaging*, 9(1):73-98, 2003.
- [12] K.V. MARDIA et P.E. JUPP, *Directional Statistics*. Wiley & Sons Ltd., 2000.
- [13] T.W. RIDLER et S. CALVARD, Picture thresholding using an iterative selection method. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(8):629-632, Août 1978.
- [14] A. ROSENFELD, *Multiresolution Image Processing and Analysis*. Springer-Verlag, 1984.
- [15] P.L. ROSIN, Thresholding for change detection. In *IEEE International Conference on Computer Vision*, pages 274-279, Bombay, Inde, Janvier 1998.
- [16] P.K. SAHOO, S. SOLTANI, A.K.C. WONG et Y.C. CHEN, A survey of thresholding techniques. *Computer Vision, Graphics and Image Processing*, 41:233-260, 1988.
- [17] J. SERRA, Morphological segmentations of colour images. In *International Symposium on Mathematical Morphology*, Paris, Avril 2005.
- [18] D. TRAVIS, *Effective Color Displays. Theory and Practice*. Academic Press, 1991.
- [19] A. TRÉMEAU, C. FERNANDEZ-MALOIGNE et P. BONTON, *Imagerie Numérique Couleur*. Dunod, 2004.
- [20] G. TZIRITAS et C. LABIT, *Motion analysis for image sequence coding*. Advances in Image Communication. Elsevier, 1994.
- [21] N. VANDENBROUCKE, *Segmentation d'images couleur par classification de pixels dans des espaces d'attributs colorimétriques adaptés. Application à l'analyse d'images de football*. Thèse de doctorat, Université des Sciences et Technologies de Lille 1, Décembre 2000.
- [22] J.Z. WANG, J. LI, R.M. GRAY et G. WIEDERHOLD, Unsupervised multiresolution segmentation for images with low depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(1):85-90, Janvier 2001.
- [23] Y. WANG, J. OSTERMANN et Y.Q. ZHANG, *Video Processing and Communications*. Prentice Hall, 2002.



Sébastien **Lefèvre**

Sébastien Lefèvre a été diplômé de l'Université de Technologie de Compiègne (Diplôme d'Ingénieur et DEA en Génie Informatique) en 1999 et de l'Université de Tours (Doctorat en Informatique) en 2002. Il est actuellement Maître de Conférences (Département Informatique / LSIIT UMR 7005 CNRS-ULP) à l'Université Louis Pasteur, Strasbourg. De 1999 à 2002, il a été ingénieur R&D au sein d'AtosOrigin. Ses thèmes de recherche concernent le traitement des images et vidéos, l'analyse et l'indexation multimédia, et la morphologie mathématique.



Nicole **Vincent**

Après des études en Mathématiques à l'Ecole Normale Supérieure, Nicole Vincent a obtenu en 1988 le grade de Docteur en Informatique à l'INSA de Lyon. Nommée Professeur à l'Université de Tours en 1996, elle est Professeur à l'Université Paris Descartes depuis 2003. Elle y dirige le Centre de Recherche en Informatique de Paris 5 (CRIP5) et l'équipe Systèmes Intelligents de Perception (SIP). Elle est spécialisée en reconnaissance de formes, traitement du signal et des images, et analyse vidéo.



