

# Un algorithme de classification couleur. Application à la reconnaissance de joueurs de football. (A color classification algorithm. Application to soccer players recognition)

N. Vandembroucke <sup>(1)</sup>, L. Macaire <sup>(2)</sup> et J.-G Postaire <sup>(2)</sup>

<sup>(1)</sup> Ecole des Mines de Douai  
Département Productique  
59508 Douai cedex, FRANCE  
email : nv@cal.univ-lille1.fr

<sup>(2)</sup> Laboratoire d'Automatique I<sup>3</sup>D  
Université des Sciences et Technologies de Lille  
Cité scientifique, Bâtiment P2  
59655 Villeneuve d'Ascq cedex, FRANCE  
email : ludovic.macaire@univ-lille1.fr

## Résumé

Dans le but d'affiner leur stratégie de jeu, les entraîneurs de football veulent disposer d'un maximum d'informations statistiques sur les matchs et les joueurs. A cette fin, nous avons proposé une méthode de suivi de joueur de football où les joueurs sont modélisés par des contours actifs après classification couleur de chaque pixel. La position de chaque joueur extrait des images d'un match peut ainsi être déterminée sauf si ce joueur est caché par un adversaire. Dans ce cas, la reconnaissance de l'équipe des joueurs par classification des pixels permet le suivi automatique de chacun d'eux.

Dans cet article, nous présentons notre algorithme de classification et nous étudions expérimentalement l'influence des paramètres de classification sur les résultats.

**Mots Clés :** Segmentation, Images couleur, Classification, Voisinage, Football.

## 1 Introduction

De nos jours, le sport de haut niveau est synonyme de gros enjeux économiques. Le football, en particulier, est devenu un support médiatique et publicitaire très largement utilisé. Le résultat d'un match a donc des conséquences très importantes et les entraîneurs veulent disposer d'un maximum d'informations statistiques sur les matchs et les joueurs dans le but d'affiner leur stratégie de jeu. A cette fin, nous avons proposé une méthode de suivi de joueurs de football à partir d'images couleurs successives acquises par une caméra fixe visant le terrain [VAN97b], [VAN97c].

D'autres auteurs se sont aussi proposés de concevoir des systèmes d'analyse d'images d'un match de football ou de rugby [GUR96], [GON95], [YOW95]. Ils utilisent directement les images retransmises par les chaînes de télévision et enregistrées sur bandes vidéo afin d'en faire l'analyse après match.

Nous avons choisi de modéliser chacun des joueurs extraits de l'image par un modèle de contour actif ou snake. Chaque

## Abstract

In order to improve their strategy, soccer coaches need accurate information about the match and the players behaviour. For this purpose, we have proposed a soccer player tracking method where the players are modeled by snakes which are evaluated after color classification of each pixel. The position of each player which is extracted from images of a match can be so determined, except if this player is hidden by another one. In this case, the recognition of the players team by pixels classification allows the automatic tracking of each of them.

In this paper, we present our classification algorithm and we experimentally study the parameters of the classification algorithm.

**Key words :** Segmentation, Color images, Classification, Neighbourhood, Soccer.

contour actif est mis à jour au fil des images dans le but de suivre automatiquement les joueurs. Or, il peut arriver qu'un joueur soit caché par un autre dans le champ de la caméra. La modélisation de chaque joueur par un contour actif n'est alors plus possible et nécessite une réinitialisation interactive. Afin d'éviter cette situation et constatant que, dans la majorité des cas, deux joueurs en contact n'appartiennent pas à la même équipe, nous avons proposé que les joueurs soient suivis indépendamment de l'équipe à laquelle ils appartiennent. Pour cela, chaque contour actif est évalué après classification couleur des pixels selon deux classes correspondant à chacune des deux équipes [VAN97a].

En effet, un opérateur humain identifie l'équipe d'un joueur par la couleur de sa tenue. C'est pourquoi nous proposons d'analyser les caractéristiques colorimétriques des joueurs qui sont extraits des images couleur à analyser. Néanmoins, les composantes classiques rouge, verte et bleue, utilisées traditionnellement pour représenter la couleur, ne sont pas forcément les plus discriminantes. Nous proposons donc de

déterminer automatiquement, parmi un ensemble de systèmes de représentation de la couleur, les trois composantes colorimétriques qui discriminent le mieux les deux équipes. Ces trois composantes forment un « espace couleur hybride ». L'algorithme de classification des pixels, qui sera détaillé ultérieurement, s'effectuera dans cet espace. Cet algorithme tient compte de la valeur colorimétrique de chaque pixel à classer mais également de celles du voisinage. Cet article a donc pour but de montrer l'influence du voisinage dans l'algorithme de classification des pixels que nous proposons.

Dans un premier temps, nous rappellerons notre méthode originale de classification de pixels. Puis, nous étudierons expérimentalement l'influence du voisinage sur les résultats de la classification. Afin de valider nos résultats, nous présentons plusieurs séries d'images à partir desquelles nous mesurons un taux d'erreur de classification dans différents cas de figure. En conclusion, nous verrons comment cette étude nous a permis d'améliorer notre algorithme.

## 2 Algorithme de classification

Nous présentons dans cette première partie une méthode de classification de pixels dans des images couleur, appliquée à la reconnaissance de joueurs au cours d'un match de football. Les différents joueurs d'un match de football se différencient principalement par la couleur de leur tenue et c'est ainsi qu'on peut les distinguer sur le terrain. Notre méthode se base donc sur un apprentissage supervisé permettant l'analyse colorimétrique des images d'un match de football en vue d'extraire les pixels qui représentent les joueurs mais également afin de déterminer un espace couleur hybride approprié au match considéré. Notre problème consiste ensuite à déterminer à laquelle des deux équipes appartiennent les pixels des images couleurs.

### 2.1 L'apprentissage supervisé

#### 2.1.1 Extraction des joueurs

L'algorithme de classification ne s'applique qu'aux pixels représentant les joueurs ; il est donc primordial de supprimer les pixels représentant la pelouse dans les images couleur à analyser. Dans un premier temps, nous proposons d'effectuer l'apprentissage des pixels représentant la pelouse afin d'en dégager ses caractéristiques. Pour cela, nous sélectionnons interactivement des « fenêtres pelouse » dans une image d'apprentissage préalablement choisie. Ce sont des fenêtres ne contenant que des pixels représentatifs de la pelouse du terrain. Elles vont nous permettre l'extraction des pixels représentant uniquement les joueurs : les « pixels joueurs ». Les histogrammes des composantes chromatiques rouge et verte normalisées, évalués dans une fenêtre pelouse, sont unimodaux. Ainsi, pour chaque mode nous pouvons déterminer deux seuils correspondant à ses limites droite et gauche. Une simple méthode de seuillage sur histogrammes dans le plan de chromaticité  $(r, g)$  nous permet ensuite d'extraire les pixels joueurs. Si la pelouse est uniforme, la sélection d'une seule fenêtre est suffisante à la segmentation des images couleurs. Par contre, si elle présente des dégradations, des ombres, des variations de

couleurs, etc., alors nous sélectionnons autant de fenêtres pelouse que celle-ci présente d'aspects différents.

#### 2.1.2 L'espace couleur hybride

Dans un second temps, nous voulons séparer les pixels selon qu'ils appartiennent aux joueurs de l'une des deux équipes, à partir de leurs composantes colorimétriques. Néanmoins, l'espace traditionnel  $(R, G, B)$  n'est peut être pas le plus discriminant. En effet, d'autres composantes colorimétriques, déduites des composantes rouge, verte et bleue, peuvent se révéler plus appropriées suivant le cas considéré. C'est pourquoi, l'analyse des pixels joueurs dans différents systèmes de représentation de la couleur nous permet de mesurer le pouvoir discriminant de plusieurs composantes colorimétriques et de déterminer ainsi un espace couleur hybride formé des trois composantes les plus discriminantes.

D'autres auteurs ont aussi tenté de comparer différents systèmes de représentation de la couleur dans le but d'extraire les composantes, les plans ou les espaces les plus appropriés à leur type de problème. Ohlander utilise neuf composantes colorimétriques afin de déterminer celle qui est la mieux adaptée à la segmentation d'images de scènes naturelles [OHL78]. Dans le même but, Ohta propose et compare à d'autres espaces de couleur un nouveau système basé sur la transformation de Karhunen-Loewe [OHT80]. Brunner comme Adel recherchent le système de représentation de la couleur le plus discriminant en vue de la détection des défauts sur des surfaces de bois [BRU92], [ADE93].

Pour constituer notre espace couleur hybride, nous retenons un ensemble de systèmes de représentation de la couleur les plus couramment utilisés, déduits du système  $(R, G, B)$  : le système de couleurs normalisées  $(r, g, b)$ , le système humain de perception de la couleur  $(I, H, S)$ , les systèmes de coordonnées couleurs de la CIE  $(X, Y, Z)$  et  $(x, y, z)$  [CIE86], les systèmes couleurs uniformes de la CIE  $(L^*, a^*, b^*)$ ,  $(L^*, u^*, v^*)$  et leurs systèmes dérivés  $(L_{ab}^*, C_{ab}^*, h_{ab}^\circ)$ ,  $(L_{uv}^*, C_{uv}^*, h_{uv}^\circ)$  et  $(L_{uv}^*, S_{uv}^*, h_{uv}^\circ)$ , le système d'Ohta  $(I1, I2, I3)$ , celui de Faugeras  $(A, C1, C2)$  [FAU79] et les systèmes de coordonnées couleurs pour les signaux de télévision  $(Y, I, Q)$  et  $(Y, U, V)$ .

Puis, nous sélectionnons interactivement pendant la phase d'apprentissage supervisé des « fenêtres joueur ». Chacune de ces fenêtres ne contient que les pixels joueurs représentant un même joueur. Pour chaque équipe, il a été sélectionné le même nombre  $N_\omega$  de fenêtres joueur. Les pixels joueurs des fenêtres joueur d'une même équipe  $j$  forment ainsi la classe de pixels joueurs  $C_j$  et nous appelons  $\omega_{i,j}$  la fenêtre joueur de rang  $i$  de l'équipe  $j$ . On note aussi  $N_c$ , le nombre de classes (ici,  $N_c = 2$ ).

Soit  $\Pi = (R, G, B, r, g, b, I, H, S, X, Y, Z, x, y, z, L^*, a^*, b^*, u^*, v^*, C_{ab}^*, h_{ab}^\circ, C_{uv}^*, h_{uv}^\circ, S_{uv}^*, I2, I3, A, C1, C2, Y', I, Q, U, V)$ , l'espace multidimensionnel défini par l'ensemble des composantes formant les systèmes couleurs présentés précédemment. Un

pixel joueur  $P(x,y)$  est représenté par un point dans l'espace  $\Pi$  dont la  $k^{\text{ième}}$  coordonnée est  $\pi^k(x,y)$ .

Soit  $N_{i,j}$ , le nombre de pixels joueurs contenus dans une fenêtre joueur  $\omega_{i,j}$ . Nous calculons, dans chaque fenêtre  $\omega_{i,j}$ , les moyennes des valeurs des pixels joueurs selon chaque composante  $\pi^k(x,y)$  :

$$m_{i,j}^k = \frac{1}{N_{i,j}} \times \sum_{P(x,y) \in \omega_{i,j}} \pi^k(x,y) \quad (1)$$

Dans le but d'évaluer le pouvoir discriminant de chacune des composantes couleur de l'espace  $\Pi$ , nous évaluons  $D_{min}^k$ , la différence interclasse minimum :

$$D_{min}^k = \min\left(|m_{i,j}^k - m_{i',j'}^k|\right) \quad (2)$$

où  $i$  et  $i'$  varient de 0 à  $N_\omega - 1$  et  $j$  et  $j'$  varient de 1 à  $N_c$  (avec  $j \neq j'$ ). De même, nous calculons  $D_{max}^k$ , la différence intraclasse maximum :

$$D_{max}^k = \max\left(|m_{i,j}^k - m_{i',j}^k|\right) \quad (3)$$

où  $i$  et  $i'$  varient de 0 à  $N_\omega - 1$  et  $j$  et varie de 1 à  $N_c$ . Finalement, le pouvoir discriminant de chacune des composantes couleur est issu du critère :

$$D^k = D_{min}^k - D_{max}^k \quad (4)$$

La valeur maximale de notre critère indique la composante chromatique la plus discriminante. Un nouvel espace couleur hybride appelé  $(\pi_1, \pi_2, \pi_3)$  réunit alors les trois composantes les plus discriminantes.

## 2.2 Classification des pixels joueurs

Après avoir extrait les pixels joueurs d'une image selon la méthode proposée précédemment, l'image couleur est ensuite convertie dans l'espace couleur hybride dans lequel est réalisée la classification.

Pour chaque équipe  $j$ , nous retenons, parmi l'ensemble des fenêtres joueur, la fenêtre joueur  $\omega_{0,j}$  qui définira une région  $RE_{0,j}$ . Cette région est formée par les pixels joueurs de la fenêtre joueur  $\omega_{0,j}$  et peut être interprétée comme une forme générique des joueurs de l'équipe  $j$ . Nous définissons aussi le centre de gravité  $CG_{0,j}$  de la région  $RE_{0,j}$  comme le centre de la fenêtre  $\omega_{0,j}$  (cf. figure [1]). On note  $N_{0,j}$ , le nombre de pixels joueurs constituant la région  $RE_{0,j}$ . C'est aussi sa surface.

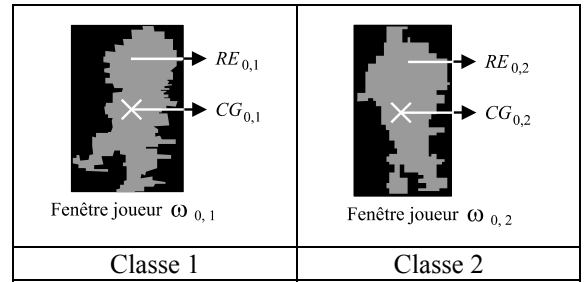


figure 1 : Fenêtres joueur retenues pour chaque classe

Soit  $RE_P$ , la région de centre  $P(x,y)$  constituée des pixels joueurs de l'image courante. Pour tout pixel joueur  $P(x,y)$  d'une image à analyser, nous évaluons la région d'intersection  $RE_I$  entre  $RE_{0,j}$  et  $RE_P$ . Pour cela, la région  $RE_{0,j}$  est translatée de sorte que son centre de gravité coïncide avec le pixel joueur  $P(x,y)$  (cf. figure [2]).  $RE_I$  est ainsi déterminée pour chaque classe  $C_j$ .

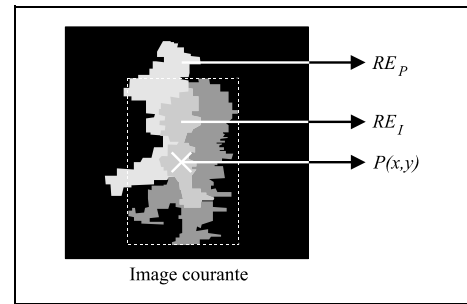


figure 2 : Région d'intersection

On note  $N_I$ , le nombre de pixels joueurs constituant la région  $RE_I$ . Nous définissons alors, dans cette région, les distances  $D_j(x,y)$  entre les moyennes des valeurs des pixels dans la région d'intersection  $RE_I$  et les moyennes  $M_j^k$  des valeurs des pixels joueurs contenus dans les fenêtres joueur  $\omega_{i,j}$  :

$$D_j(x,y) = \frac{1}{3} \times \sum_{k=1}^{k=3} \left( M_j^k - \sum_{P(x,y) \in RE_I} \frac{\pi_k(x,y)}{N_I} \right) \quad (5)$$

avec :

$$M_j^k = \frac{1}{N_\omega} \times \sum_{i=0}^{N_\omega-1} m_{i,j}^k \quad (6)$$

Chaque pixel joueur  $P(x,y)$  est ensuite assigné à la classe  $C_j$ , pour laquelle  $D_j(x,y)$  est minimum.

L'algorithme de classification que nous proposons prend en considération les composantes colorimétriques du pixel joueur à classer mais également celles des pixels joueurs voisins. Les pixels joueurs voisins sont ceux qui appartiennent à la région d'intersection  $RE_I$ . Cette région, déterminée pour chaque pixel joueur et pour chaque classe  $C_j$ , joue donc un rôle très important dans la classification. De plus, elle est différente selon le voisinage du pixel joueur traité et selon la région  $RE_{0,j}$ . Il semble donc

intéressant d'étudier dans quelle mesure la région  $RE_{0,j}$  influence les résultats de la classification.

### 3 Influence du voisinage sur la classification

La région  $RE_{0,j}$ , représentative de chaque classe, est caractérisée par deux paramètres : sa forme et sa taille. Afin de déterminer dans quels cas de figure ces deux paramètres engendrent les meilleurs résultats sur la classification, nous devons les faire évoluer sur différentes images et mesurer le taux d'erreur de classification dans chaque cas. Nous allons tout d'abord présenter différentes images à partir desquelles nous allons expérimenter notre approche. Ces images sont acquises à partir de matchs différents. Pour chaque match, nous réalisons l'apprentissage supervisé décrit dans la partie 2.1 afin de pouvoir extraire les pixels joueurs et déterminer l'espace couleur hybride. Nous proposons

ensuite une méthode pour mesurer le taux d'erreur de classification afin de comparer les différents résultats de la classification. Enfin, nous étudierons, dans un premier temps, l'influence de la forme de la région choisie puis, dans un second temps, la taille de cette même région. Pour cela, nous nous placerons dans différents cas de figure et nous mesurerons le taux d'erreur de classification afin de déterminer quelle est la meilleure configuration.

#### 3.1 Images analysées

Nous proposons de réaliser notre étude expérimentale dans trois cas correspondant à trois matchs différents. Pour chacun des cas, nous avons choisi interactivement quatre images (a), (b), (c) et (d) (cf. figures [3], [4] et [5]) dans lesquelles il s'agit d'extraire et de classer les pixels joueurs entre deux classes.

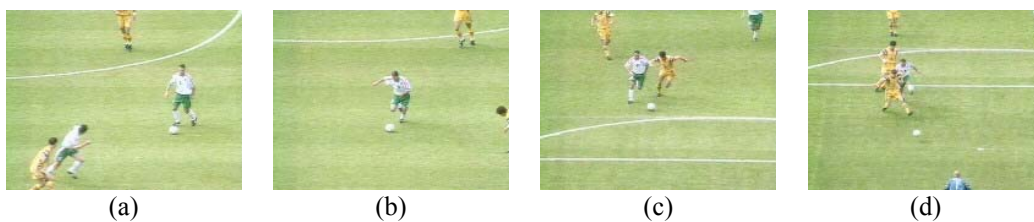


figure 3 : Images extraites du match 1 (cas n° 1)

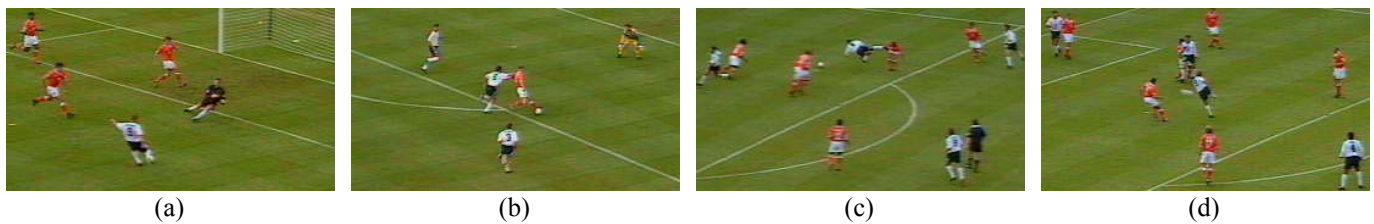


figure 4 : Images extraites du match 2 (cas n° 2)



figure 5 : Images extraites du match 3 (cas n° 3)

Un apprentissage supervisé nous a permis d'extraire les pixels joueurs et de sélectionner interactivement 3, 7 et 5 fenêtres joueur par classe respectivement pour les trois cas considérés (cf. figure [6], [7] et [8]) :

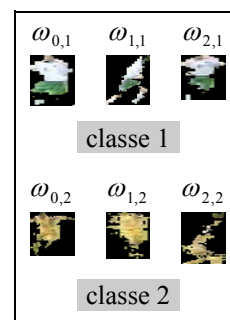


figure 6 : Fenêtres joueur sélectionnées pour le cas n° 1

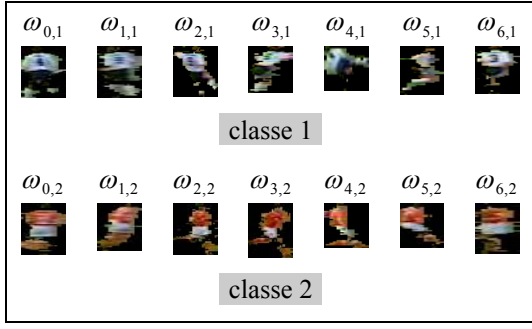


figure 7 : Fenêtres joueur sélectionnées pour le cas n° 2

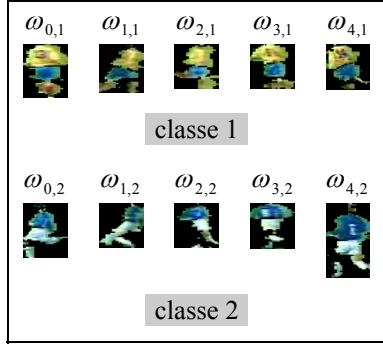


figure 8 : Fenêtres joueur sélectionnées pour le cas n° 3

Dans chacun des trois cas, nous avons évalué l'espace couleur hybride  $(\pi_1, \pi_2, \pi_3)$  correspondant selon la méthode présentée dans partie 2.1.2 (cf. tableau [1]).

	cas n° 1	cas n° 2	cas n° 3
Espace Couleur Hybride	$(H, B, Z)$	$(I, U, u^*)$	$(H, B, I2)$

tableau 1 : Espace hybride déterminé pour chaque cas

### 3.2 Mesure du taux d'erreur de classification

Afin de comparer les résultats de la classification dans différents cas de figure, nous proposons une méthode de mesure du taux d'erreur de classification dans les images couleur. Cette méthode est basée sur la comparaison des images dont les pixels joueurs ont été classés par notre algorithme avec des images dont nous avons fixé interactivement la classe de chaque pixel. Il nous faut donc connaître a priori à quelle classe appartiennent les pixels joueurs de l'image à analyser. Ainsi, pour chacune des 4 images des trois cas, nous avons interactivement déterminé à quelle classe appartient chaque pixel joueur pour former l'image binaire idéale des joueurs de l'équipe 1,  $IMA_1'$  et l'image binaire idéale des joueurs de l'équipe 2,  $IMA_2'$ . Soient  $N_1'$  et  $N_2'$ , le nombre de pixels joueurs respectifs présents dans ces images.

Après application de notre algorithme de classification des pixels de l'image, nous obtenons aussi deux images binaires dans lesquelles sont représentées respectivement les pixels joueurs assignés à la classe  $C_1$  et ceux assignés à la classe  $C_2$ . Nous avons appelé  $IMA_1$  et  $IMA_2$  ces images résultats. Il nous faut donc comparer  $IMA_1$  à  $IMA_1'$  et  $IMA_2$  à  $IMA_2'$ . Pour cela nous opérons un ET logique

respectivement entre ces images afin de mettre en évidence uniquement les pixels appartenant effectivement à chacune des deux classes. Il ne reste plus alors qu'à dénombrer les pixels résultant de cette opération et à comparer leurs nombres  $N_1$  et  $N_2$  respectivement à  $N_1'$  et  $N_2'$ .

Dans le cas où  $N_c = 2$ , les pixels représentant les arbitres ou les gardiens de but n'appartiennent a priori à aucune des deux classes de pixels joueurs. Ils ne figurent donc pas dans les images  $IMA_1'$  et  $IMA_2'$ . Or, pendant le processus de classification il seront a fortiori assignés à l'une des deux classes et figureront dans les images  $IMA_1$  ou  $IMA_2$ . Grâce à l'opération ET logique, il ne seront pas pris en compte dans le résultat et ne seront pas comptabilisés pour évaluer le taux d'erreur de classification car on ne considère ici que l'erreur relative aux deux classes.

Le taux d'erreur de classification  $\varepsilon$  qui correspond à l'assignation d'un pixel joueur de la classe  $C_1$  ( $C_2$ ) à la classe  $C_2$  ( $C_1$ ) est alors donné par la relation suivante :

$$\varepsilon = \left( 1 - \frac{N_1 + N_2}{N_1' + N_2'} \right) \times 100 \quad (7)$$

Prenons comme exemple l'image (d) de la figure [5]. On y distingue trois joueurs, un gardien de but et le ballon. Pour cette image, nous avons formé les images  $IMA_1'$  et  $IMA_2'$  (cf. figure [9]) et nous avons relevé le nombre de pixels joueurs idéal :  $N_1' = 608$  et  $N_2' = 1416$ .



figure 9 : Création des images idéales

Après application de notre algorithme de classification, nous obtenons les images  $IMA_1$  et  $IMA_2$  de la figure [10].



figure 10 : Résultat de l'algorithme de classification

Nous réalisons alors le ET logique entre  $IMA_1$  et  $IMA_1'$  et entre  $IMA_2$  et  $IMA_2'$  (cf. figure [11]). Nous obtenons  $N_1 = 571$  et  $N_2 = 1411$ .

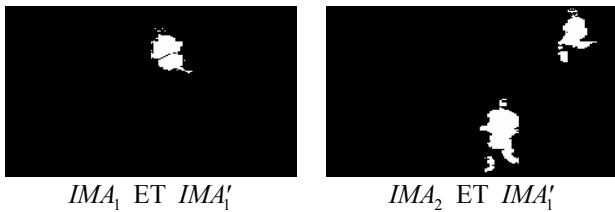


figure 11 : Résultat du ET logique

Le taux d'erreur de classification est alors pour cette image :  $\varepsilon = 2,08\%$  .

### 3.3 Influence de la forme de la région représentative de chaque classe

Nous allons maintenant étudier l'influence de la région  $RE_{0,j}$  choisie. Comme l'illustre la figure [1], nous choisissons deux fenêtres joueur différentes  $\omega_{0,1}$  et  $\omega_{0,2}$  parmi l'ensemble des fenêtres joueur pour représenter les joueurs de chacune des deux équipes. Les pixels joueurs de

ces fenêtres joueur forment ainsi respectivement les régions  $RE_{0,1}$  et  $RE_{0,2}$  dont la forme et la surface caractérisent respectivement chacune des deux équipes. Nous mesurons alors les taux d'erreur de classification obtenus dans ces conditions (cf. tableau [2]).

	(a)	(b)	(c)	(d)	moyenne
cas n° 1	4,53	0,18	8,35	10,42	<b>5,87</b>
cas n° 2	0	1,04	4,66	8,96	<b>3,67</b>
cas n° 3	0,67	1,90	0,24	2,08	<b>1,22</b>

tableau 2 : Taux d'erreur avec  $RE_{0,j}$

Nous obtenons les images suivantes dans chacun de nos trois cas (cf. figure [12], [13], [14]). Sur ces images les pixels joueurs assignés à la classe  $C_1$  sont représentés en bleu (foncé) et ceux assignés à la classe  $C_2$  sont représentés en vert (clair).



figure 12 : Résultats de la classification avec une région  $RE_{0,j}$  pour le cas n° 1



figure 13 : Résultats de la classification avec une région  $RE_{0,j}$  pour le cas n° 2



figure 14 : Résultats de la classification avec une région  $RE_{0,j}$  pour le cas n° 3

Or, nous ne pouvons pas affirmer qu'un joueur présente une forme bien déterminée à cause de son aspect déformable, non plus que des joueurs d'équipe différente ont des formes différentes. Au contraire, un joueur est une entité à part entière qu'il appartienne à une équipe ou à une autre. C'est pourquoi, nous proposons de représenter l'ensemble des joueurs d'un match par une forme unique et indépendante de l'équipe à laquelle il appartient afin de ne pas privilégier une classe plutôt qu'une autre. De plus, les joueurs ne prenant pas tous la même forme sur une image et étant donné qu'ils changent de forme à chaque image à cause de

leur mouvement, nous choisissons de représenter tous les joueurs par une région type  $RT_{0,j}$  de forme rectangulaire pleine dont on suppose qu'elle généralise l'ensemble des formes que peuvent prendre les joueurs.

Afin de comparer les résultats de la classification sur l'ensemble des images présentées, nous choisissons une région rectangulaire dont la surface est égale à la surface de la région  $RE_{0,j}$  avec le même rapport hauteur / largeur.

Pour cela, on note  $LR_{0,j}$ , la largeur de la région  $RE_{0,j}$  et  $HR_{0,j}$ , sa hauteur. Ces paramètres, ainsi que  $N_{0,j}$ ,

dépendent, en fait, des conditions d'acquisition des images. On note aussi  $LR'_{0,j}$ , la largeur de la région type  $RT_{0,j}$  équivalente et  $HR'_{0,j}$ , sa hauteur. Nous avons alors les relations suivantes :

$$\frac{HR_{0,j}}{LR_{0,j}} = \frac{HR'_{0,j}}{LR'_{0,j}} \quad (8)$$

$$N_{0,j} = HR'_{0,j} \times LR'_{0,j} \quad (9)$$

Et, nous obtenons :

$$HR'_{0,j} = \sqrt{N_{0,j} \times \frac{HR_{0,j}}{LR_{0,j}}} \quad (10)$$

$$LR'_{0,j} = \sqrt{N_{0,j} \times \frac{LR_{0,j}}{HR_{0,j}}} \quad (11)$$

Les tableaux [3], [4] et [5] donnent les paramètres associés à chaque fenêtre joueur  $\omega_{0,j}$  respectivement pour les trois cas considérés :

	Région $RE_{0,j}$			Région $RT_{0,j}$	
	$LR_{0,j}$	$HR_{0,j}$	$N_{0,j}$	$LR'_{0,j}$	$HR'_{0,j}$
$\omega_{0,1}$	19	41	393	13	29
$\omega_{0,2}$	27	34	336	16	21

tableau 3 : Taille des régions dans le cas n° 1

	Région $RE_{0,j}$			Région $RT_{0,j}$	
	$LR_{0,j}$	$HR_{0,j}$	$N_{0,j}$	$LR'_{0,j}$	$HR'_{0,j}$
$\omega_{0,1}$	24	42	475	16	29
$\omega_{0,2}$	20	42	434	14	30

tableau 4 : Taille des régions dans le cas n° 2

	Région $RE_{0,j}$			Région $RT_{0,j}$	
	$LR_{0,j}$	$HR_{0,j}$	$N_{0,j}$	$LR'_{0,j}$	$HR'_{0,j}$
$\omega_{0,1}$	29	43	712	22	32
$\omega_{0,2}$	32	44	516	19	27

tableau 5 : Taille des régions dans le cas n° 3

En procédant à la classification des pixels dans les quatre images des trois cas présentés avec une région type  $RT_{0,j}$  de forme rectangulaire et de taille  $LR'_{0,j} \times HR'_{0,j}$ , nous obtenons des taux d'erreur très proches mais supérieur de ceux du tableau [2] (cf. tableau [6]).

	(a)	(b)	(c)	(d)	moyenne
cas n° 1	3,74	4,46	13,12	8,43	<b>7,44</b>
cas n° 2	0,65	1,28	4,10	9,26	<b>3,83</b>
cas n° 3	0,82	3,73	0,42	2,62	<b>1,90</b>

tableau 6 : Taux d'erreur avec  $RT_{0,j}$

La région  $RT_{0,j}$  dépend de la fenêtre joueur  $\omega_{0,j}$  choisie par l'opérateur au cours de la phase d'apprentissage supervisé. Les résultats des tableaux [6] et [2] sont donc très dépendant de ce choix. Il en résulte, en outre, que les régions  $RT_{0,j}$  peuvent être de tailles différentes alors qu'elles doivent toutes représenter la même entité, c'est à dire des joueurs.

Ceci nous a donc conduit à utiliser des régions types identiques pour les deux classes. Notre algorithme n'opère plus alors qu'avec une seule région quelque soit la classe à déterminer. Les taux d'erreur présentés dans le tableau [7] correspondent au cas où la région  $RT_{0,2}$  est égale à la région  $RT_{0,1}$  déterminée précédemment. Nous appelons  $RT$ , cette région.

	(a)	(b)	(c)	(d)	moyenne
cas n° 1	3,45	1,25	12,11	8,32	<b>6,28</b>
cas n° 2	0,65	1,28	4,70	9,88	<b>4,13</b>
cas n° 3	0,35	2,45	1,03	2,12	<b>1,49</b>

tableau 7 : Taux d'erreur avec  $RT$

Au regard des résultats précédents, il apparaît que, statistiquement, le choix d'une région type de forme rectangulaire n'accentue le taux d'erreur que de 0,5 % environs. En revanche, le gain en nombre d'opérations est de plus de 50 %. En effet, si on utilise une seule région pour représenter l'ensemble des joueurs au lieu d'une région par classe, alors une seule région d'intersection sera évaluée à chaque itération. De plus, la fenêtre dans laquelle la région est comprise est pleine et rectangulaire, ce qui nous permet de supprimer les tests d'appartenance d'un pixel joueur à la région (tous les pixels joueurs de la fenêtre appartenant à cette région).

#### 3.4. Influence de la surface de la région type, représentative des classes

Nous proposons maintenant d'étudier l'influence de la surface de la région type rectangulaire  $RT$ , identique pour toute les classes. On note  $LR'$ , sa largeur et  $HR'$ , sa hauteur. Nous faisons alors évoluer  $LR'$  de 1 à 40.  $HR'$  évolue proportionnellement avec le rapport moyen hauteur/largeur, notée *rapport*, de toutes les fenêtres joueur  $\omega_{i,j}$  sélectionnées au cours de l'apprentissage supervisé :

$$HR' = \text{rapport} \times LR' \quad (12)$$

avec :

$$\text{rapport} = \frac{1}{N_\omega \times N_c} \times \sum_{i=0}^{N_\omega-1} \sum_{j=1}^{N_c} \frac{HR_{i,j}}{LR_{i,j}} \quad (13)$$

Où  $HR_{i,j}$  est la hauteur de la région  $RE_{i,j}$  constituée des pixels joueurs de la fenêtre joueur  $\omega_{i,j}$  et  $LR_{i,j}$ , sa largeur.

A chaque incrémentation de  $LR'$ , nous mesurons le taux d'erreur de classification  $\varepsilon$  dans chacune des images de nos trois cas. Les figures [15], [16] et [17] montrent l'évolution du taux d'erreur de classification en fonction de la taille de la région  $RT$  dans chacune des quatre images

des cas respectifs n° 1, n° 2 et n° 3. Sur ces figures, la courbe en rouge montre donc l'évolution  $\varepsilon = f(LR')$  pour l'image (a), celle en vert montre la même évolution pour l'image (b), celle en bleu pour l'image (c) et la courbe noire est utilisée pour l'image (d).

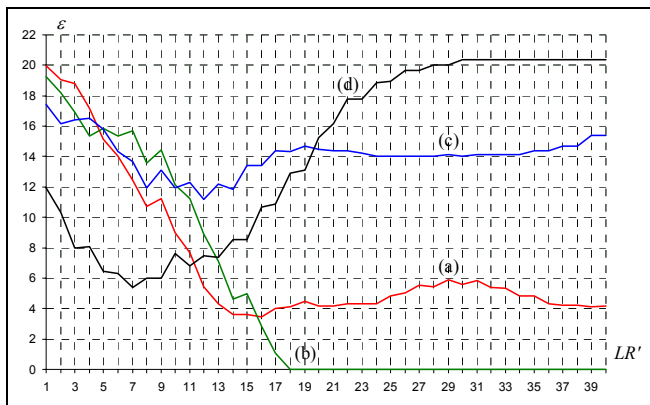


figure 15 : Evolution  $\varepsilon = f(LR')$  dans le cas n° 1

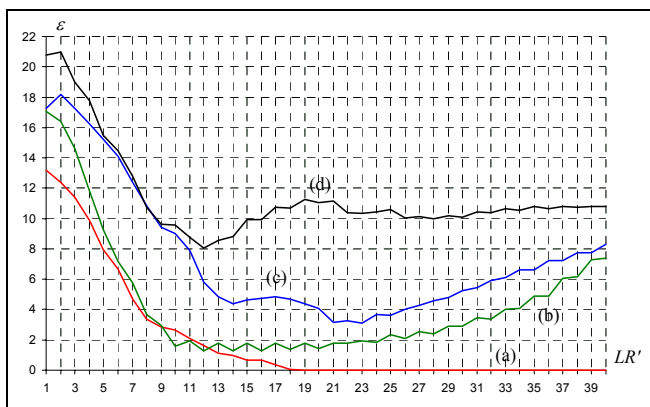


figure 16 : Evolution  $\varepsilon = f(LR')$  dans le cas n° 2

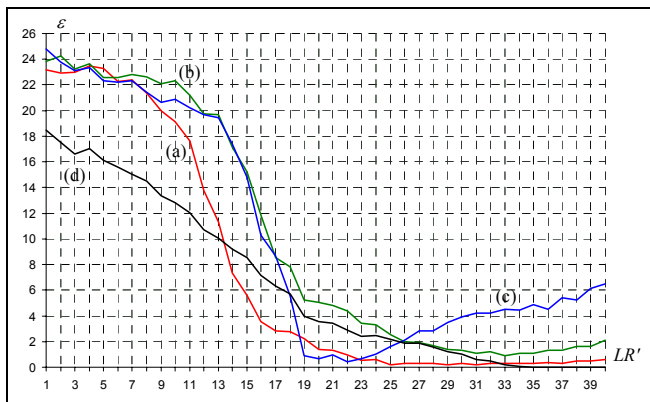


figure 17 : Evolution  $\varepsilon = f(LR')$  dans le cas n° 3

En analysant les courbes précédentes, nous constatons qu'il peut y avoir un écart d'erreur important entre les images d'une même séquence. Ceci s'explique par le fait que les joueurs sont soit éloignés les uns des autres dans l'image, soit très proches ou en contact. En effet, supposons qu'à la périphérie d'un pixel joueur traité, les joueurs soient éloignés les uns des autres. Pour classer le pixel joueur considéré, la région  $RE_i$  ne contiendra alors que les pixels joueurs représentant le même joueur. Au contraire quand les joueurs sont très proches les uns des autres, la région  $RE_i$  contiendra alors des pixels joueurs pouvant représenter des joueurs d'une autre équipe. Ce phénomène

est d'autant plus accentué que  $LR'$ , la largeur de la région avec laquelle est évaluée l'intersection, est élevée.

Afin de dégager une tendance générale, nous présentons sur les figures [18], [19] et [20], l'évolution du taux d'erreur de classification moyen  $\varepsilon_{moyen}$  sur l'ensemble des quatre images des trois cas considérés.

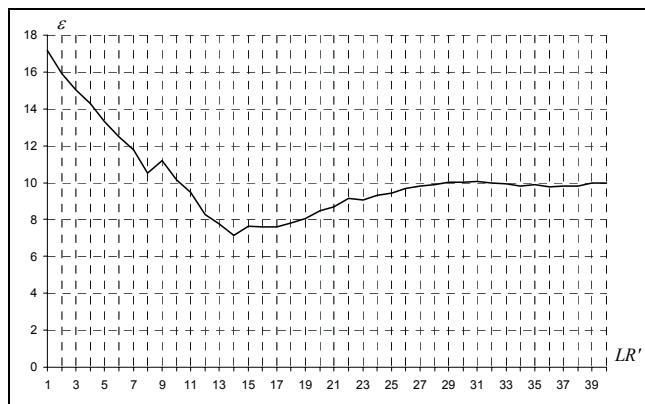


figure 18 : Evolution  $\varepsilon_{moyen} = f(LR')$  dans le cas n° 1

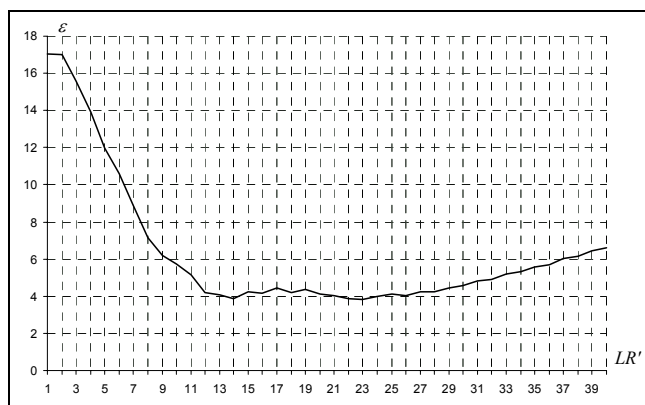


figure 19 : Evolution  $\varepsilon_{moyen} = f(LR')$  dans le cas n° 2

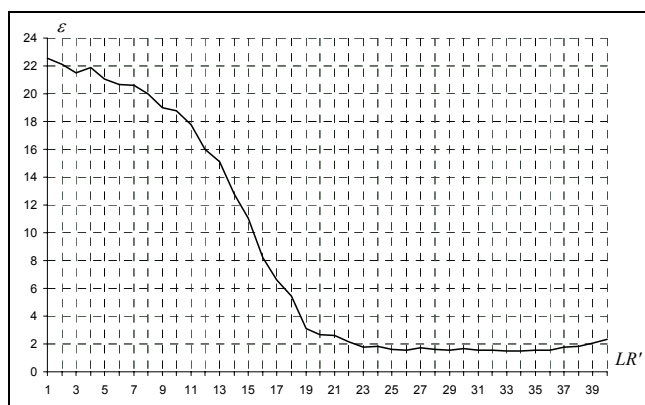


figure 20 : Evolution  $\varepsilon_{moyen} = f(LR')$  dans le cas n° 3

Au regard de ces courbes, il apparaît que l'efficacité de l'algorithme augmente rapidement au fur et à mesure que la taille de la région augmente jusqu'à atteindre un taux d'erreur minimum. Puis, cette efficacité diminue progressivement lorsqu'on atteint des tailles trop importantes. En effet, si la surface de  $RT$  est faible, la région d'intersection  $RE_i$  ne contiendra pas assez de pixels joueurs d'un même joueur et si, au contraire, la surface de  $RT$  est trop importante, alors  $RE_i$  risque de contenir des pixels joueurs d'un autre joueur.

Il semble que les résultats soient satisfaisants lorsqu'on choisit une région rectangulaire dont la surface correspond approximativement à la surface moyenne, notée *surface*, d'un joueur dans les images traitées :

$$surface = \frac{1}{N_\omega \times N_c} \times \sum_{i=0}^{N_\omega-1} \sum_{j=1}^{N_c} N_{i,j} \quad (14)$$

Ce cas de figure correspond à une région *RT* de largeur  $LR'_{moyen}$  et de hauteur  $HR'_{moyen}$  définies par les équations suivantes :

$$\frac{HR'_{moyen}}{LR'_{moyen}} = rapport \quad (15)$$

$$HR'_{moyen} \times LR'_{moyen} = surface \quad (16)$$

D'où :

$$LR'_{moyen} = \sqrt{\frac{surface}{rapport}} \quad (17)$$

$$HR'_{moyen} = \sqrt{surface \times rapport} \quad (18)$$

On remarque alors que statistiquement,  $LR'_{moyen}$  et  $HR'_{moyen}$  correspondent à un taux d'erreur de classification  $\epsilon_{moyen}$  très proche du taux minimum, notée  $\epsilon_{min}$ , dans chacun des cas (cf. tableau [8]).

	$LR'_{moyen}$	$HR'_{moyen}$	$\epsilon_{moyen}$	$\epsilon_{min}$
cas n° 1	14	24	<b>7,15</b>	<b>7,15</b>
cas n° 2	15	26	<b>4,24</b>	<b>3,87</b>
cas n° 3	20	26	<b>2,66</b>	<b>1,47</b>

tableau 8 : Taux d'erreur moyen et minimum

## 4 Conclusion et perspectives

Aux vues des résultats de cette étude, nous proposons de modifier notre algorithme simplement en représentant tous les joueurs par une région type rectangulaire appelée *RT*. La surface de cette région est déterminée en fonction du rapport moyen entre la hauteur et la largeur des fenêtres joueur  $\omega_{i,j}$  sélectionnées et en fonction du nombre moyen de pixels joueurs contenus dans ces mêmes fenêtres joueur (cf. équation (12) à (18)). Dans ces conditions, nous obtenons les résultats des figures [21], [22] et [23] qui restent très semblables à ceux des figures [12], [13] et [14]. En revanche, le nombre d'opérations est réduit de plus de 50 %. Cependant, nous remarquons que les erreurs de classification se produisent au niveau des chaussettes des joueurs. Dans cette zone, beaucoup moins de pixels joueurs voisins interviennent dans la classification. On observe d'autre part sur l'image (c) de la figure [21] que le joueur en haut à droite est identifié comme appartenant à l'équipe des joueurs en blanc et vert alors qu'il appartient, en fait à l'équipe des joueurs en jaune. Cette erreur est due, ici, au choix de l'espace couleur hybride mal adapté à cette image. Cet espace dépend des fenêtres joueur sélectionnées pendant la phase d'apprentissage supervisé qui, pour ce cas, sont en moins grand nombre que pour les deux autres cas. On voit donc que notre approche dépend aussi du nombre  $N_\omega$  de fenêtre joueur  $\omega_{i,j}$  sélectionnées qui doit être suffisamment important pour obtenir l'apprentissage le plus représentatif possible. D'autre part, notre critère de sélection de l'espace couleur hybride ne tient pas compte de la corrélation qui existe entre les composantes qui le forment ni de la dimension optimale de l'espace hybride. Nos travaux se dirigent actuellement dans ce sens afin de pouvoir distinguer au mieux non seulement les joueurs des deux équipes mais aussi les deux gardiens de but et les arbitres.



figure 21 : Résultats de la classification avec la région *RT* pour le cas n° 1

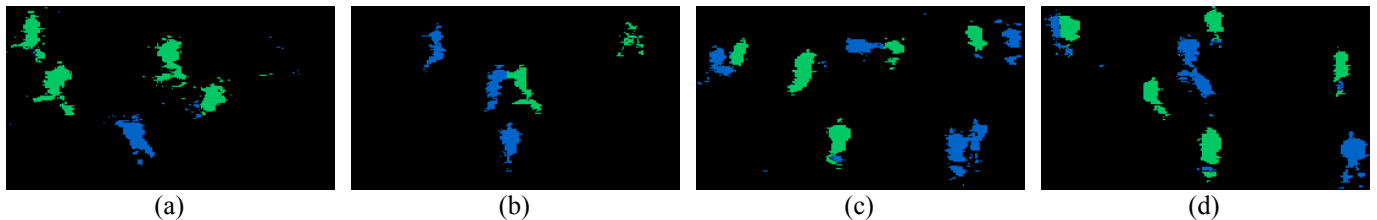


figure 22 : Résultats de la classification avec la région *RT* pour le cas n° 2



figure 23 : Résultats de la classification avec la région  $RT$  pour le cas n° 3

## 5 Références bibliographiques

[VAN97c], N. VANDENBROUCKE, L. MACAIRE, C. VIEREN et J.-G. POSTAIRE, « Contribution of a color classification to soccer players tracking with snakes », *IEEE International Conference on System, Man and Cybernetics*, Orlando, volume 4, 1997, pages 3660-3665.

[VAN97b], N. VANDENBROUCKE, L. MACAIRE et C. VIEREN, « Suivi automatique de personnes en mouvement par analyse d'images couleurs successives. Application au suivi de joueurs de football », *Seizième colloque GRETSI sur le Traitement du Signal et des Images*, Grenoble, volume 2, 1997, pages 917-920.

[VAN97a], N. VANDENBROUCKE, L. MACAIRE et J.-G. POSTAIRE, « Soccer player recognition by pixels classification in an hybrid color space », *Multispectral and Hyperspectral Imagery III (SPIE proceedings)*, volume 3071, Orlando, 1997, pages 23-33.

[GUR96], P. GURDJOS, P. DALLE, et S. CASTAN, « Tracking 3D coplanar points in the invariant perspective coordinates plane », *13th International Conference on Pattern Recognition*, Vienne, 1996, pages 493-497.

[GON95], Y. H. GONG, H. C. CHUA et T. S. LIM, « An automatic video parser for TV soccer games », *Second Asian Conference on Computer Vision*, Singapore, 1995, pages 509-513.

[YOW95], D. YOW, B. L. YEO, M. YEUNG et B. LIU, « Analysis and presentation of soccer highlights from

digital video », *Second Asian Conference on Computer Vision*, Singapore, 1995, pages 499-503.

[OHL78], R. OHLANDER, K. PRICE, et D. R. REDDY", « Picture segmentation using a recursive region splitting method », *Computer Graphics and Image Processing*, 1978, volume 8, pages 313-333.

[OHT80], Y. I. OHTA, T. KANADE, et T. SAKAI, « Color information for region segmentation », *Computer Graphics and Image Processing*, 1980, volume 13, pages 222-241.

[BRU92], C.-C. BRUNNER, A. G. MARISTANY, D. A. BUTLER, D. VANLEUWEEN et J. W. FUNCK, « An evaluation of color spaces for detecting defects in Douglas-fir veneer », *Industrial Metrology*, 1992, volume 2, n° 3-4, pages 169-184.

[ADE93], M. ADEL, D. WOLF, R. HUSSON et R. VOGRIG, « Evaluation of color spaces in computer vision : an application of wood defects detection », *IEEE International Conference on Systems Man and Cybernetics*, volume 2, Le Touquet, 1993, pages 499-504.

[CIE86], Comission International de l'Eclairage, « Colorimetry, 2nd Edition », 1986, n° 15.2.

[FAU79], O. D. FAUGERAS, « Digital color image processing within the framework of a human visual model », *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1979, volume 27, n° 4, pages 380-393.