UNIVERSITÉ DE NANTES

Polytech'Nantes
École d'ingénieurs de l'université de Nantes

## STAGE DE MASTER 2R

### SYSTÈMES ELECTRONIQUES ET GÉNIE ELECTRIQUE

### PARCOURS SYSTÈMES ELECTRONIQUES

ANNÉE UNIVERSITAIRE 2007/2008

# Importance Map Computation of HD Video For The Coding In H.264/AVC

Xiaole Fang

Ecole Polytechnique de l'Université de Nantes

Laboratoire IRCCYN

*Encadrant du stage :*

*Vincent RICORDEL and Olivier BROUARD*

Stage effectué du (05/02/08) au (05/07/08)

# Abstract

The latest standard for the HDTV is H.264/AVC, and because of the great resolution of HDTV (High Defined TV), the encoding process is computationally very expensive and is not suitable in a real-time context. The ArchiPEG project plan to pre-process the HD video before the encoding, A segmentation based on motion, texture, color and so on has been already achieved for the Group of Frames (GOF). In this report, according to the Human Visual System and some results of other researchers, we generate a new method of the Importance Map (IM) calculation by analyzing the color, size, position and motion elements of the video frame in H.264 coding, and this IM is necessary for setting up the quantization step of the encoder. For the experiment result, a comparison with eye tracking data are done.

**Keywords**: video coding, HDTV, H,264/AVC, Importance Map, color, size, position, motion

# Resume

La dernière norme pour la TVHD est le standard H.264/AVC, et en raison de la grande résolution de la TVHD (TV Haute Définition), le processus d'encodage est très coûteuse en temps de calcul et est difficilement implémentable  en temps réel.  Le projet ArchiPEG prévoit de réaliser une  pré-traiter de la vidéo HD avant le codage, une segmentation spatio-remporelle pour groupe d'image  basée sur la motion, la texture, la couleur ainsi que a déjà été réalisé. Dans ce rapport, d'après les connaissances sur le théorie du système visuel humain et des résultats d'autres chercheurs, nous générons une nouvelle méthode de l'importance Plan (IM) calcul en analysant la couleur, taille, la position et le mouvement des éléments de l'image vidéo en codage H.264 , Et ce IM est nécessaire de mettre en place l'étape de quantification de l'encodeur. Pour l'expérience, une comparaison avec des yeux de suivi des données sont effectuées.

**Mots-clés:** H, 264/AVC, Carte importance, la couleur, taille, position, le mouvement

# Contents

# 1 Introduction

## 1.1 Our laboratory

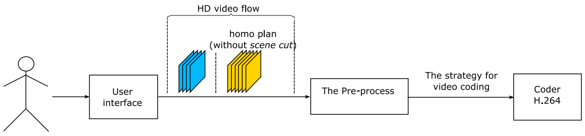
Our lab is the "Institut de Recherche en Communications et Cybernétique de Nantes" (IRCCyN), which is a joint research unit (UMR 6597) of CNRS " Center National de la Recherche Scientifique". It is associated with two university institutions in Nantes: "Ecole Centrale de Nantes", "Université de Nantes" and with "Ecole des Mines de Nantes" .

IRCCyN is composed of about 200 persons: 87 permanent researchers (14 from the CNRS, the others have a teaching position in one of its associated institutions), 18 engineers, technicians and administrative employees, and 95 persons with temporary positions (mainly PhD students, visiting professors and Post-Docs) [1].

IRCCyN's research objectives concern the development of fundamental, methodological, and technological research in the fields of cybernetics, with a particular emphasis with mechanical systems and cognitive psychology. There are several research groups of IRCCyN they are active in: automatic control, signal and images processing, video communications, robotics, design methods in mechanics, embedded real-time systems, modelling and optimization of production process, logistic and production systems, discrete event systems, cognitive psychology and ergonomics. A longstanding and strong contractual activity links IRCCyN with large industrial partners. Our lab is also taking part in several National and European research projects and networks of Excellence. IRCCyN is a partner in 5 national competitive poles "Pôles de compétitivité" : "EMC2", "Image et réseau", "System@tic", "Prevensor and Automobile Haut de Gamme". It is developing strong international cooperation strategy. The institute is a partner in formal international research agreements with Mexico, the Czech Republic, China, Korea, Poland, Italy, South Africa, Malaysia, USA and Russia. IRCCyN plays an essential role in the postgraduate studies (Masters and PhD) of its associated institutions [1].

IRCCyN's team is a very warm, very enthusiastic and very serious-minded organization, the team munbers all have good technique, they have done many contributions for the scientific research. Study in a such good condition, I feel so happy, and because of that, I have learned so much useful skill for my development.

Figure 1.1: The Architecture



## 1.2 The context of study

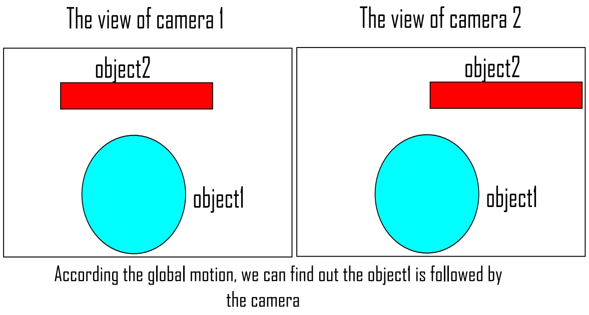
Nowadays, the video has been became an indispensable part in our life, and in the video using, on line video is a very important point. In our study, we will give a solution for some facets of the HDTV (High Definition TV video coding), which supports $1920 \times 1080$ pixels as a frame definition in progressive mode with a frame rate of 50Hz.

### 1.2.1 The video compression standard and the whole structure for study

When we have to face HDTV coding , we want to reduce its information cost, but we also want to make the display be good, it will be used a technique, which is video compression. How is the video compression? As Iain E.G. Richardson said [2], that means reducing the information which we will send. It is a very complex course of events, and it has been a subject developed for many years. During the time, people has created many many compression standards, such like MPEG-1, MPEG-2, and so on. Among them, MPEG-2 is a very classic standard, which has been used as a base for digital TV and DVD-Video for nearly 10 years old, in these moment, MPEG-2 has proved its effectiveness. However, the HDTV coding demand higher quality for the video, the MPEG-2 seems not suitable for that. A new standard named H.264/MPEG-4 AVC has been developed by the Joint Video Team of ISO/IEC MPEG and ITU-T Video Coding Expert Group aims at allowing a bit rate reduction of 50% compare to MPEG-2 at the same quality, it is more efficient, and the advance is get from the multi prediction modes, multi reference frames and higher motion vector resolution. However, the reduction of the rate is obtained by an increasing of the time for computing [4], so that makes it difficultly real time applications. To add a pre-process part for the H.264 video compression standard is a strategy for it.

The video transmits as a frame form, they will firstly do some pre-process, which aim to calculate some relative information for the compression processing, and then these frames will be sent to H.264 PROCESS to achieve the compression, at the end, the video information will be compressed as code to send to the channel, this is our whole cause of events for the HDTV code.

Figure 1.2: Global motion


The view of camera 1
The view of camera 2
object2
object1
object2
object1
According the global motion, we can find out the object1 is followed by the camera

### 1.2.2 The ArchiPEG project: the pre-process

In the Figure 1.1, the whole structure of the video coding has been described, after the HD video flow, we will do the pre-process, so how is the pre-process? Usually the pre-process job is to segment the frame, because separating objects from the picture help us handle the H.264 process, there are many ways to achieve that. So for segmenting the frame, in the previously work, Vincent, Oliver and others [4], they focus on the calculation of the global motion. Because once you can get the way of the global motion, you will know how the camera moving (the global motion is the camera shot motion), like shown in Figure 1.2, we will find out which object in the screen the camera want to focus on, so the global motion estimation (GME) methods are effective for segmentation and tracking of the video objects.

In the Vincent, Oliver and others paper [4] a multi-resolution motion estimation has been proposed in order to accelerate the processing of the HDTV coding, the frames can be shrink and it will reduce the calculation. The frames are spatially filtered and sub-sampled by a factor of six, that means the original 1920 × 1080 pixels per frame will become 640 × 360 pixels. In the HDTV, they define the value of FPS (frame per second) should be 50, that means 0.02 second per picture, so using 9 frames as a group to segment has been decided, because it will 9 frames cost 0.18 second, it is very near 0.2 second, which is defined as a perfect point for human visual. They call the group as GOF, show as Figure 1.3. Also a special tool named tube, the macro block or object is tracked with this tube in the GOF and in it the motion is uniform, this tracking is depending on the minimal MSEg, which is mean the global error. Here they use the 5th (center) frame as the reference, and calculate a coefficient which bases on the three YUV components of block named MSE between reference one and the others, sum them, it will deduce the MSEg.

Figure 1.3: The GOF and the tube

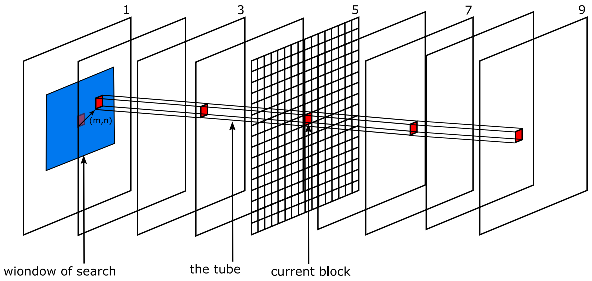
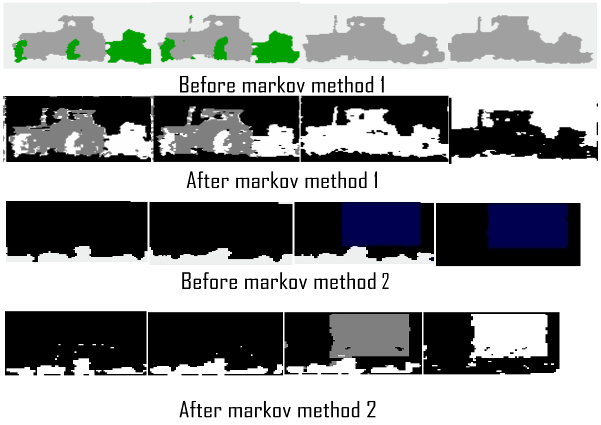


Figure 1.4: The comparing of the frame with Markov method and the one without


Before markov method 1

After markov method 1

Before markov method 2

After markov method 2

The chosen motion vector in a block has the lowest MSEg between the current block and its corresponding blocks (shown in Figure 1.3, the No.1, 3, 7 and 9 frames). And meanwhile the motion vector are estimated at the lowest resolution (the 640 × 360 pixels), and then, they are up-scaled appropriately to the higher resolution (the 1920 × 1080 pixels) to be used in an initial search point. And then an affine model has been used [4], the affine model has used 4 parameters to represent a motion vector, these 4 parameters will be collected in a histogram and be counted, and which has the most times appearing in the histogram, is the global motion parameters. So according that the global motion which means the camera moving can be deduced. Last they give different objects relative labels for identification in the objects.

After it has still not finished the segmentation yet, because only the above segmentation, it is easy to generate the distortion in the bonders of object and the frame. So Vincent, Oliver and others also use a MARKOV method [5] to solve the problem of the segmentation, they used the features like spatial, color, texture and so on, combine them to lead a global energy potential to make sure the tracking is right (shown in figure 1.4).

By finishing these works, the segmentation is completed.

## 1.3 Conclusion

During this chapter, the context of our research has been introduced. In the ArchiPEG project, we have already segmented the objects from the frame very precisely and our final goal is to give a sensible solution to the HDTV coding which use the H.264 standard, so giving a research to H.264 standards is very necessary. And at the same time, the next step we have to use the information that we get from the previous work, we should solve the segmentation information to generate the parameters for the encoder. We will think about the visual attention for the frame, give some studies of the color, size of the object, position of the object and so on to use the result of the segmentation, we call these feature influence Importance Map [9] (IM). If we can find out the relation between the attraction of object and the IM, we can code the video much better. Next chapter we have to study the processing of the H.264 encoder, and according it and the segmentation result, find a point to begin our research on IM.

# 2 The Bibliography for the study

In this chapter, we will begin our discussion on the H.264 processing and optimized algorithm, we will find our goal in this discussion, and then mention some already existed optimized methods for our goal's facet, last analyze them.

## 2.1 The Architecture of H.264 encoder

The encoder of H.264 is shown as Figure 2.1:

In the encoder, the frame will split to different kinds of macro blocks according to the condition, and the encoder will use the macro block as a unit to do processing.

The encoder supports two models for the prediction, they are intra and inter models, and the encoder will choose one as the way to predict the frame. If it chooses the inter model prediction, in the block ME (Motion Estimate) in the Figure 2.1, the current block will firstly compare with the reference frame which comes from some frames that has been processed before, and then obtain the motion estimation; next in the block MC (Motion Compensation) the encoder will use the motion estimation to give the reference frame compensation and deduces a special frame, sends this frame to P, last the current frame will reduce the special frame, and the encoder use this residual part named $D_n$ as the transmitting information. And if the intra prediction has been chosen, then the reconstructed frame $F_n^{'}$ will be used, the reconstructed frame $F_n^{'}$ is generated by the processing frame at the last times, it is not the origin frame, and H.264 has support several kinds of spatial intra prediction modes (the inter an intra prediction modes are shown in Figure 2.2), according the condition of the current frame, people can choose a most suitable mode for the prediction in the block Choose Intra Prediction, and the choosing is depending on how important the block is. Then the frame will use the spacial prediction to deduce a particular frame in the block Intra Prediction and send it to P point, last the special frame will be reduced by current frame just as the inter prediction.

After the prediction, the residual part $D_n$, is sent to do the DCT (Discrete Cosine Transform) transform in T block and get DCT coefficients, then the coefficients will be quantified by the specific quantization step in block Q, and then it will go in two ways,

Figure 2.1: The H.264 encoder

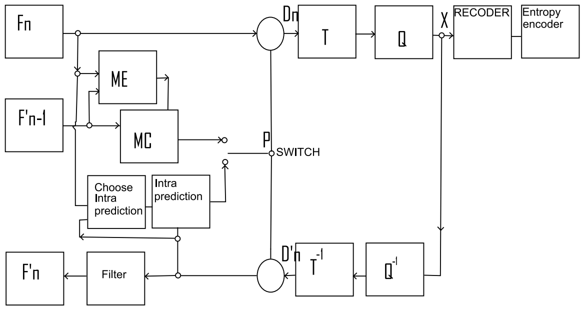
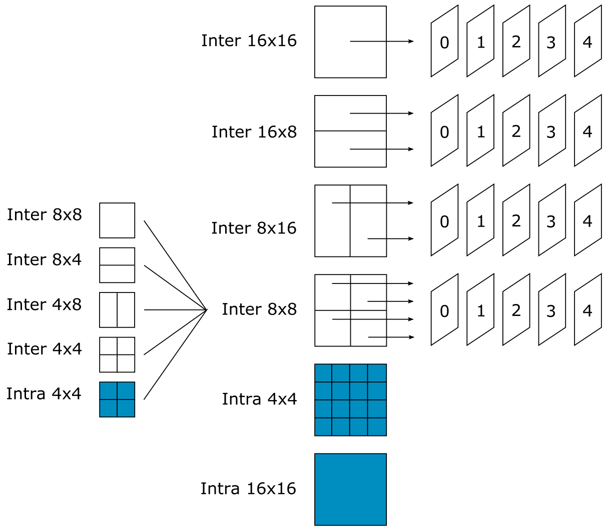


Figure 2.2: The prediction modes for inter and intra



one way is to record and use the head information to generate the code in the block Recorder and Entropy Encoder; another way is to do the reverse quantization and DCT, to get the $D_n^{'}$, and then use reference frame to generate the frame $F_n^{'}$, this frame may become the next reference frame.

Before this H.264 encoder, we have already done the pre-process, and the objects in the frame can be identified with the different labels, according to this, we want to use some rules about the object's feature to decide how important of the object, and then give a corresponding application of prediction to it. So in a word, in this paper, our goal is to find the IM for the objects, and then use it to help the prediction and quantization. And at this point, some people have already shown their ideas for us, next, we will describe them and do analyzing for some of them, and we will try to find the best way according to our specific situation.

## 2.2 A perceptually region method for Important Maps based on the HVS

No matter what people will use to optimize the algorithm, they should think about an important point: HVS (Human Visual System). The Human Visual System Model, often referred to as the Human Visual System (HVS), is used by image processing, video processing and Computer vision experts to deal with biological and psychological processes that are not yet fully understood. The model is used to simplify the behaviors of what is a very complex system. As our knowledge of the true Human Visual System improves, the model is updated. HVS shows the study of the human eye and perception's response to the views and the signals. It can be concluded that there are several points will cause the influences to attracting human attention, like color, textures and so on. Then we will see how the watchers already did before in this research [7].

Wilfried Osberger, Anthony J.Maeder and others in their papers [9, 11, 10, 8] has mentioned a synthetic algorithm which is based on the perception from human of the objects, so they compute the object IM, here we call their method as Osberger's method.

Osberger's method is basing on HVS (Human Visual System), and the conditions are a little similar to us, their agreements also come from segmentation, although the unit for segment is different (ours is GOF, theirs is frame). From the paper, it is said that after a segmentation, we can get the object's color contrast, size, shape, location and background, and combine these 5 features, it can deduce an IM that will influence the attraction of the objects (shown as Figure 2.3). And how do the features works? Osberger's method has given the answer as below:

1. Contrast: it mentions that the Grey level for the object is the higher than its neighboring objects, then the IM for the contrast is the greater, a method has been used as:
$$I_{contrast}(R_i) = \overline{gl}(R_i) - \overline{gl}(R_{i-neighbours}),$$

where $\overline{gl}(R_i)$ is the mean Grey level of region $R_i$, and $\overline{gl}(R_{i-neighbours})$ is the mean Grey-level of all of the neighboring regions of $R_i$.

2. Size: the size has an important effect in attracting attention, the larger regions are more likely to attract the attention than smaller ones, but a saturation point exists, after this point the IM due to the size levels down, so they deduce this:

$$I_{size}(R_i) = max(\frac{A(R_i)}{A_{max}}, 1.0),$$

where $A(R_i)$ is the area of region $R_i$ in pixels, and $A_{max}$ is a constant used to prevent excessive weighting being given to very large regions.

3. Shape: in the paper, it tells that the longer and thinner shapes have found to be visual attractors, so the IM calculation for the shape as:

$$I_{shape}(R_i) = \frac{bp(R_i)^{sp}}{A(R_i)},$$

where $bp(R_i)$ is the number of pixels in the region $R_i$ which border with other regions, and sp is a constant.

4. Location: an idea is that usually the eyes are directed at the center 25% of a screen is shown in the paper, so if the object has included this center section the more, then the IM for the location is the bigger, the equation as:

$$I_{location}(R_i) = \frac{centre(R_i)}{A(R_i)},$$

where $center(R_i)$ is the number of pixels in region $R_i$ which are also in the center 25% of the image.

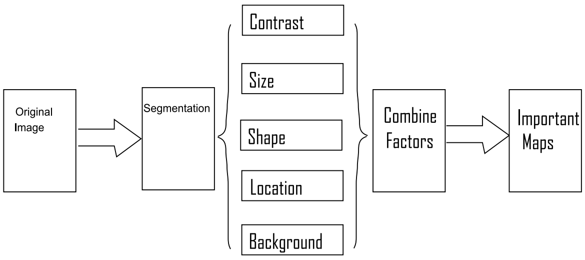
5. Background: people's view will more likely to be attracted to the objects in the foreground than those in the background, so to find out which object is the foreground and which is the background, the IM is given by:

$$I_{bg}(R_i) = 1.0 - max(\frac{borderpix(R_i)}{0.5 \times tot\_borderpix}, 1.0),$$

where $borderpix(R_i)$ is the number of pixels in region $R_i$ which also border on the image, and $tot\_borderpix$ is the total number of image border pixels.

According these 5 points, they combine them with a sum calculation to generate the IM for the segmentation.

Figure 2.3: The features deduced by segmentation



The IM for the object are not only from the segmentation, in Osberger's method, the another two points: motion vector and activity (texture).

In the calculation of the IM for the motion, they propose two thresholds with a minimal and a maximal value , so that the motion vector of the object has 5 different rules:

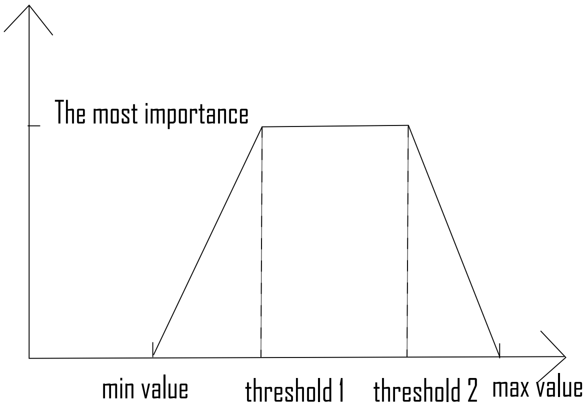
1. if it is below the minimal value, it means the IM for the motion is 0;
2. if it is between the minimal value and threshold, it mean the IM for the motion will raise as the motion vector raise;
3. if it is between smaller and bigger threshold, that means the IM will keep a highest value;
4. if it is between the bigger threshold and maximal value, it shows that the bigger the motion value the smaller the IM;
5. last stage, if it is bigger than the maximal value, the IM will keep a 0 value.

The situation is like Figure 2.4 shows.

And the calculation of the IM is in this form:

$$ImpMot_j = \begin{cases} 0.0, if\ mot_j < mot_{min} \\ \frac{mot_j - mot_{min}}{mot_{pl} - mot_{min}}, if\ mot_{min} < mot_j < mot_{p1} \\ 1.0, if\ mot_{pl} < mot_j < mot_{p2} \\ \frac{mot_{max} - mot_j}{mot_{max} - mot_{p2}}, if\ mot_{p2} < mot_{_j} < mot_{max} \\ 0.0, if\ mot_j > mot_{max} \end{cases},$$

Figure 2.4: The graph of the motion importance



where $mot_j$ is the magnitude of the motion vector for block j, $mot_{min}$ is the minimum important motion parameter (set to 0.0 $deg/sec$), $mot_{p1}$ and $mot_{p2}$ are peak motion importance parameters (both set to 10.0 $deg/sec$), and $mot_{max}$ is the threshold for maximum important motion (set to 20.0 $deg/sec$). High importance is therefore assigned to regions undergoing medium to high motion, while areas of low motion and areas undergoing very high motion (i.e. untrackable motion) are assigned low motion importance.

For the IM of the textures, it splits the condition of the texture in 2 different representations: one is flat, another one is edge or texture. They give different contributions according the different representations.

At the end of Osberger's method, these 3 IM have been combined, use some normalized ways, which is like this form:

$$N = \frac{imp + 2.0 \times imp}{2.0 \times imp + imp},$$

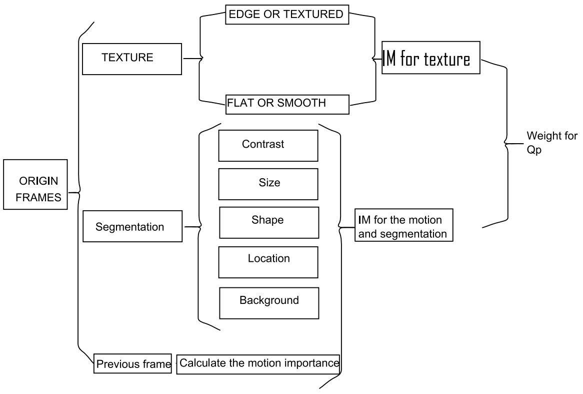
and we get $N1$ for texture and $N2$ for segmentation and motion, and we can get our MQUANT[10]:

$$MQUANT = N1 \times N2 \times Q$$

(the $Q$ means Q step) [3]. The whole processing is just like Figure 2.5 shows:

Osberger's method has shown us a very clear way to deduce the IM from the features, and the structure of the Osberger's is very good and easy to understand, although our condition is to use GOF as a unit, not the same as the frame unit in Osberger's method, we can still use their structure to analyze our feature IM.

Figure 2.5: The Osberger's method structure



## 2.3 Different algorithms for the features

Because the Osberger's method has shown us a very sensible structure, so in our research, we will use their way as a basement. However, the method is combined by the different algorithms of the features. And although Osberger's method has already mentioned their ideas for the features from their situation, our condition is different, so we can do some improve from the origin Osberger's method, we may update the algorithms for some features, or add more features. So in this part, we will analyze other feature methods based on the structure of the Osberger's method. After the GOF segmentation, we can get some features like: color, size and so on, so let us see how other researchers give the idea of them.

### 2.3.1 Color contrast

Color contrast is a important feature of an object. In Zaheer Aziz and Mertsching's paper [3], it gave a way to deduce the IM from the features, which is fast and robust to generate the Map for the object, we call their way as Zaheer Aziz's method. It also give some points for generating the IM, so we can see how do they do in color contrast this point.

Before the method show up, a factor of region perimeter has been introduced, which called $f_i^p$, and it will gain a high value for regions with moderate size and a small value for very small (noise), a very large (background) regions. Then a formula is created like $f_i^p$:

$$f_i^p = k^P_{scale} \frac{(P^i_R - k^P_{min} P_I)(P_I - P^i_R - k^P_{max} P_I)}{(P_I/2 + k^P_{min} P_I - k^P_{max} P_I)},$$

where $P_I$ is the perimeter of the input image, $P_R$ is the perimeter of the bounding rectangle of $R_i$, $k^p_{min}$ is the minimum percentage of the image perimeter below which region should be neglected, $k^p_{max}$ is the maximum percentage of the image above which a region should be regarded as background, and $k^p_{scale}$ is a scaling constant to bring the highest value obtained from the rest of the expression equals to 1.

Then the feature work begins, the first feature mentioned is color contrast, here it uses the below 8 points to analyze the color contrast:

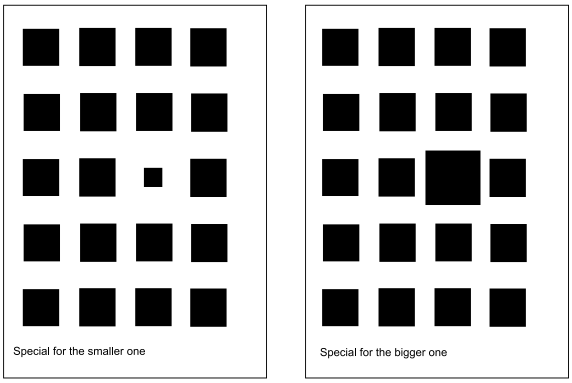
1. Contrast of Saturation: except the condition that low saturated region is surrounded by highly saturated one, highly saturated colors tend to attract attention.
2. Contrast of Intensity: except the condition that low intensity (dark) region is surrounded by high intensity (bright) one, bright colors better to catch the eye.
3. Contrast of Hue: the difference of hue angles on the color wheel contributes to creation of contrast, the distance the higher the contrast will be the more obvious.
4. Contrast of Opponents: if the colors reside on the opposite sides of the hue circle, it will course a high amount of contrast, and the colors residing in the first half of the hue circle, it will be treated as the active colors, it will be more attractive.
5. Contrast of Warm and Cool: colors present in the first 45 degree of the hue circle is the warm color, and the warm and cold colors can create a strong contrast in which warm colors remain dominant.
6. Accent Colors: if an object covering a large area of the scene will become trivial for attention, and in another side, the object in a small relative area, will offer the contrast, these colors called accent colors, and they are dominate.
7. Dominance of Warm Colors: there is a condition that whatever the contrast exists, the warm colors will attract attention always.
8. Dominance of Brightness and Saturation: highly bright and saturated colors will be keep their activity regardless of their hue values.

Obviously this feature explain is making more sense than the one of Osberger's method, it nearly thinks about all the states for the color contrast.

### 2.3.2 Size

About the size, In Zaheer Aziz and Mertsching's paper[3], it also gives a way to generate IM for this feature. It have mentioned two examples to explain the two scenarios where the only obvious feature to determine salience is the size of objects and the uniquely sized objects are the obvious attractors of attention (Figure 2.6). It is said that the perimeter factor $f^p_i$ contributes by suppressing large sized background regions and unnoticeable small sized regions and the rest comes from the contrast in size with respect to the neighborhood. So they use a voting style mechanism similar to the before color IM detection into the exclusiveness of size with respect to the

Figure 2.6: Different situations for the size exclusiveness



neighborhood and the global context. And then the conditions for the IM calculation is listed:

1. If the condition of the two objects is one surrounded by another one, and at the same time, the size of surrounded one is at a constant level according the size of the surrounding one, that will have a big importance.
2. If there is no surrounding situation, and the objects' size ratio is in a constant scale we will deduce a normal contrast, and it will not give a very big importance.
3. If there is not the condition above, that will cause little contrast, nearly no importance.

So from the list above, the equation for the size votes can be deduced:

$$\begin{cases} 1 & when\, R_j \odot R_i \, and \, \alpha(R_i)/\alpha(R_j) \leq k_1^a \\ 0.5 & when\, \alpha(R_i)/\alpha(R_i) \leq k_1^a \\ 0.5 & when\, \alpha(R_i)/\alpha(R_j) \geq k_2^a \\ 0 & otherewise \end{cases},$$

where $k_1^a$ and $k_2^a$ are thresholds constants and $\odot$ means surrounding. After this, the tool used to calculate, is just similar to the color contrast calculation method[3], the contribution from $R_j \in \eta_i$ and those from the context is shown as:

$$\begin{cases} X^i = \sum_{j=1}^{p_i} V_{ij}^s & \forall R_J \in \eta_i \\ Y^i = \sum_{l=1}^{n} V_{ij}^s/2 & \forall R_l \in R, l \neq i \end{cases},$$

At last a IM for size has been given:

$$S_a^i = \frac{X+Y}{p_i + (n-1)} S^{max} f_i^p,$$

where $S^{max}$ is the maximum amount of salience value that can be assigned to a region due to a single feature, $p_i$ is the count of neighbors in the neighborhood list of objects, $n$ is the count of regions in $R$.

Also compare with the size calculation of the Osberger's method, the Zaheer Aziz's method seems better, it does not only think about the how the size change, it also give the ideas about the surrounding condition.

### 2.3.3 Symmetry, orientation and eccentricity

Symmetry, orientation and eccentricity these features are not mentioned by Osberger's method, but in the Zaheer Aziz's method, it has show us the calculation of them very particularly. First of all, they analyze the character of these features; and then they define the contribution of region $R_j$ to the IM of $R_i$ in terms of orientation, eccentricity and symmetry as $v_{ij}^o$, $v_{ij}^e$ and $v_{ij}^s$, and use the characters to calculate them [3]; last, use the tool that introduced before to generate the IM for them: $S_o^i = \sum_{j=1}^{p_i} v_{ij}^o$ for the orientation IM, $S_e^i = \sum_{j=1}^{p_i} v_{ij}^e$ for the eccentricity and $S_s^i = \sum_{j=1}^{s_i} v_{ij}^s$ for the symmetry.

It has a reason to think about these features, however, in our situation, we do not think it is suitable. Because they are not the obvious features that come from segmentation straightly, and it will cause much more computing for find if these features existed.

### 2.3.4 Human face location

Although Human face has not mentioned by Osberger's method, it is a very important attractive point in the image, so let us see how some researchers do the study for this part. Chai and Bouzerdoum's paper [6] represented an idea for the face focus, they believe that in the frame, if there are faces existed, they are very attractive, so they give 5 points to get the face's localization. The points are:

1. Color segmentation: that is to use a proposed universal skin color map to classify pixels of the input image into skin-color and not-skin-color.
2. Density regularization: it will examine the bitmap produced by the color segmentation in previous stage. It attempts to highlight regions of the bitmap that have higher probability belong to a face and to remove regions that have lower probability.
3. Luminance regularization: this is another process that uses a different cue value to further remove skin-color pixels that do not belong to the face.

4. Geometric correction: a horizontal and vertical scanning is performed in order to identify and subsequently remove the presence of any odd structure in the bitmap produced by the third point.
5. Contour extraction: the output bitmap from fourth point is converted back from block to pixel resolution, and this point is achieved by utilizing the edge information that is already made available by the color segmentation in the first point.

The paper thinks that these five points can be used to face localization. In our condition, we are different from them, we have already accomplished the segmentation, and all the discussing of the paper [6] is depending on the image condition, not the video, so it may be not necessary for us to use such many points to locate the face position.

### 2.3.5 The motion

For a video, motion is a very important element. In the paper of Tang, Chen and others [12], they give a very special way to calculate the IM from motion, they use a way to normalize the motion vector, define $(mvx_{nij}, mvy_{nij})$ (here $ij$ means the motion location, and $n$ uses as a label) is the motion vector, they first find the maximal value of $\sqrt{mvx_{nij}^2 + mvy_{nij}^2}$, set it as $I_{max}$, and the intensity of the motion intensity of location $(i, j)$ is $I_{nij} = \sqrt{mvx_{nij}^2 + mvy_{nij}^2}/I_{max}$. Next they do some research in two facets of motion, one is spatial consistency of the directions of the motion vectors, the calculation is in this way:

$$Cs_{nij} = -\sum_{b=1}^{n_s} ps_n(b)log(ps_n(b)),$$

where $ps_n(b)$ is the probability distribution function which represents the direction in a block window (as example $16 \times 16$ pixels macro block), so if in the macro block most of its region is belonging to a same object, so the value of the $Cs_{nij}$ will be small like Figure 2.7; another one is temporal consistency of directions of the motion vectors:

$$Ct_{nij} = -\sum_{b=1}^{n_s} pt_n(b)log(pt_n(b)),$$

they use L frames, give their direction probability distribution function $pt_n(b)$, so that if in L frames the number of direction change in a object the less the $Ct_{nij}$ the smaller because of the equation of the temporal consistency of directions of the motion vectors (Shown in Figure 2.8). At last it combines the $I, Cs, Ct$ together to get the IM for motion. Even though this is a very nice tool to computing, but it is not suitable in our situation, because we have already finished segmentation as GOF, we can get all the object's motion in the GOF.

Figure 2.7: 16 × 16 macro block direction

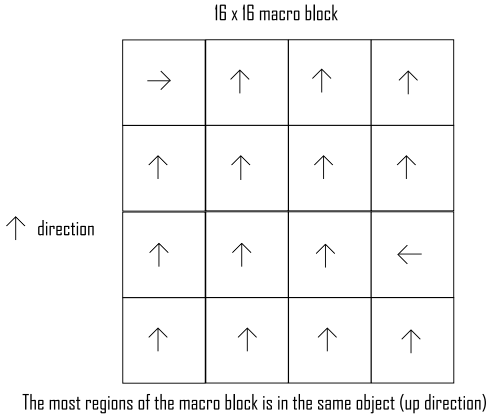
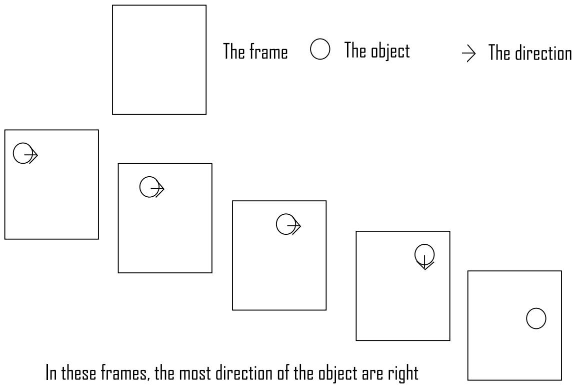

16 x 16 macro block
direction
The most regions of the macro block is in the same object (up direction)

Figure 2.8: L frames for the object's direction



## 2.4 Conclusion

According to the studying in H.264 encoder and other some people's methods in the IM, there are somethings can be concluded.

First of all, these IM calculations are all from the segmentation, in the before job, the segmentation was finished, and we also get the many features of the objects like color, size, location and so on.

Secondly, in which level the object attract people's attention, how to judge the attraction of the different objects is the most important point of IM calculation, so the HVS is a very important basis for the IM, the HVS defines the features contribution for IM, we should follow it.

Last, the IM calculation should be according the corresponding condition of them, like the different ways to segment or others.

In my opinion, How to make the method adaptive to our condition is the main problem of our research, our work is depending on the GOF unit segmentation, the methods we talk before are not all useful to us. Except that, our situation also can deduce some features that the before method can not contain, like span life for the objects, so we will try to do some study in this facet.

# 3 The element's calculation of IM

In the last chapter, we have already seen several methods for the IM computing, and here we will begin our calculation of the IM for video coding.

First of all, we should think about the structure of the IM elements generation, Wilfried Osberger, Anthony J.Maeder's paper [9, 11, 10, 8] have supported us a very good structure, it is very clear that every aspect's function has been expressed, so we will use this structure. And then which point should be chosen is the next question, in Zaheer Aziz and Mertsching's paper [3], the Color Map calculation has been given in detail, and also in HVS system, color do play an important part, so we will take Color Map as one point; Size Map has been mentioned in Wilfried Osberger's paper and Zaheer Aziz's paper, it is also an indispensable element for the IM, so Size Map will be the second element we choose; then according to the HVS, in a picture, the objects in different positions will give different levels attraction, so Position Map will also be my one chosen; go to the last, all the above three points are for the Image Importance, however, our situation is to compute the IM for video coding, so we will think the elements of the video, and most important point of them is the motion, and Motion Map is the last component that we choose. We will use Color, Size, Position and Motion. These 4 Maps, with the structure of the Osberger's paper representing to deduce the IM, next we will begin our job for each element's computing (Shown in Figure 3.1).

## 3.1 Color Map

In our calculation of Color Map, all the works are based on the HSV platform, because that HSV system will provide us hue, saturation and value (intensity) these 3 aspects of the picture, it makes the analyzing easier, so our first job is to transform the video frame to form of HSV (shown in Figure 3.2).

Because the video compression is using the macro block as the unit, here we define the macro block is $16 \times 16$ pixels, so with the macro block splitting the frame, the original resolution will divided by $16 \times 16$, that means in the video frame $16 \times 16$ pixels will become only 1 pixel, so this 1 pixel will represent the character of the $16 \times 16$ pixels, here we will compute the average value of the hue, saturation and intensity in the macro block, and use it to be the characters of this 1 pixels, the result frame will be shown in Figure 3.3.

Figure 3.1: The structure

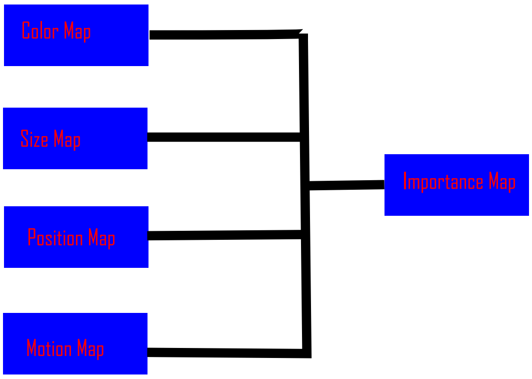


Figure 3.2: RGB form (left) to HSV form (right)

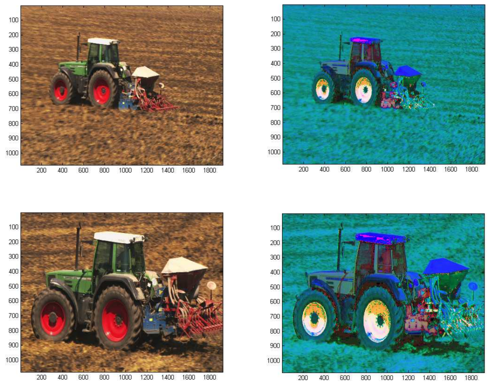


Figure 3.3: Original resolution (above) to the resolution of after macro block cutting (below)

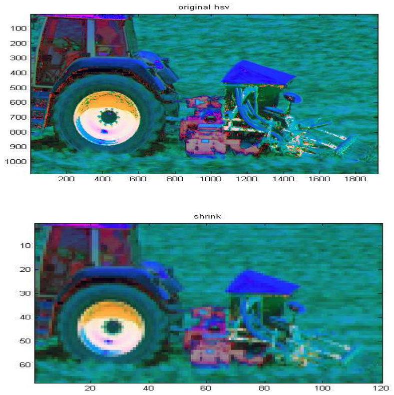
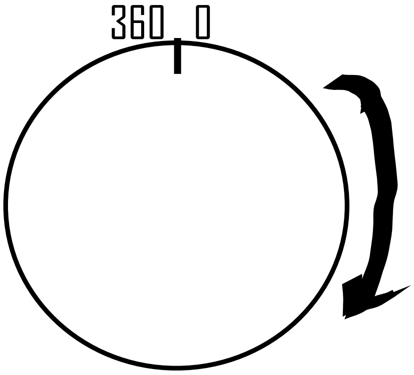


Figure 3.4: The Hue distribution



In Zaheer Aziz and Mertsching's paper [3], the Color Map is represent by the contrast of the hue, saturation and intensity, and that is what we will do next, find out the contrast of the each elements.

### 3.1.1 Hue contrast for Color Map

Here we will compute the contrast of the hue level, as the Zaheer Aziz and Mertsching's paper [3] said, if the difference of the two object's hue value get the higher, and then the attraction will be the higher, in our condition that we will not use the hue value of the objects, we choose to take the hue value of every macro block.

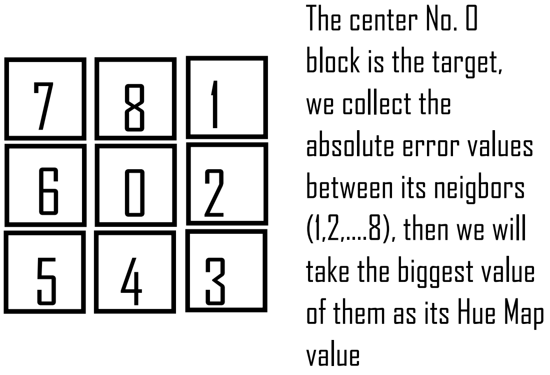
Firstly of all, the absolute difference values will be calculated between the target macro block and its neighbors, however, hue value is circle distribution, it is just as shown in Figure 3.4, it will loop the color, as hue value varies from 0 to 360 degree, the resulting color varies from red through yellow, green, cyan, blue, and magenta, and returns to red, so the biggest distance value will be 180 degree. So we will handle the calculation like this: if the we get the absolute error which from the hue value of target block subtract the neighbor is bigger than 180, we will use the 360 reduce by it, and if it is smaller than 180, then we will keep it. Then we keep all these values in a group, last we will compare the values in the group, and pick the greatest one as the Hue Map value for the macro block (as shown in Figure 3.5).

The process is that:

$$HueMap = Max_{i=1}^{8}(|hue_0 - hue_i|),$$

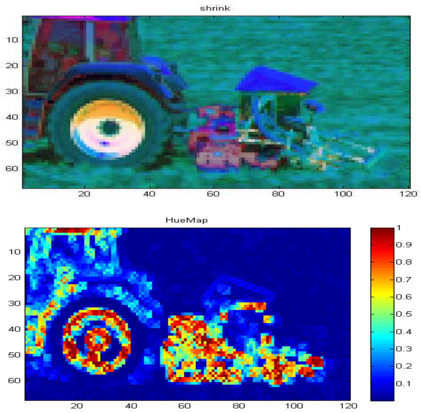
After calculating the HueMaps of all the macro blocks, we will do some normalizing to make the value scale between 0 and 1, firstly we will treat all these value in macro block position as a matrix, then we find out the biggest value in this matrix, last we will use the matrix divides by this biggest value, here is the expression:

Figure 3.5: The Hue Map calculation



The center No. 0 block is the target, we collect the absolute error values between its neigbors (1,2,....8), then we will take the biggest value of them as its Hue Map value

Figure 3.6: the shrink HSV picture (above) and HueMap (below) picture



$$HueMap = HueMap_{origin}/maxvalue(HueMap_{origin}),$$

And example of Hue Map is shown as in Figure 3.6.

### 3.1.2 Saturation contrast and Intensity contrast for Color Map

Saturation is an element that can display the color state, when the value of saturation is 0, the colors are unsaturated (i.e., shades of gray), when the value is 1, the colors are fully saturated (i.e., they contain no white component).

Intensity (Value) is an element which can indicate the brightness, as the intensity varies from 0 to 1, the brightness increases.

Figure 3.7: the shrink HSV (above) and the Saturation Map (below)

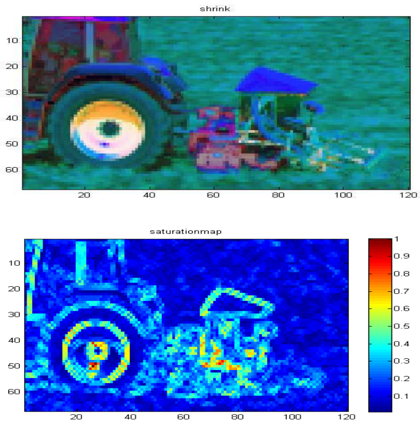
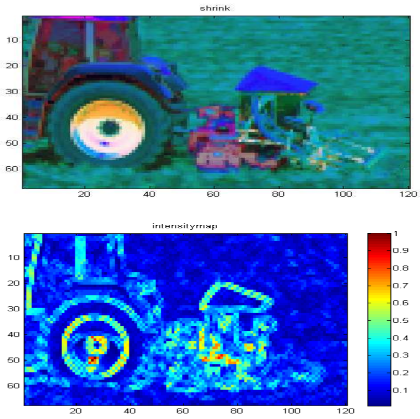


Figure 3.8: the shrink HSV (above) and the Intensity Map (below)



We will calculate the Saturation Map and Intensity Map with the same way as the Hue Map computing except the loop part judgment. So the expression will be

$$SaturationMap = Max_{i=1}^{8}(|saturation_0 - saturation_i|),$$

and

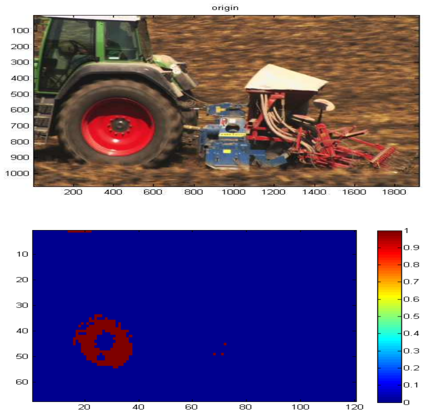
$$IntensityMap = Max_{i=1}^{8}(|intensity_0 - intensity_i|),$$

The same examples for Saturation Map and Intensity Map (Figure 3.7 and Figure 3.8).

### 3.1.3 The warm color for Color Map

In Zaheer Aziz and Mertsching's paper [3], it has mentioned a concept of warm and cool color that colors present in the first 45 degree of the hue circle is the warm color (red, yellow....), whatever the contrast exists, the warm colors will attract attention

Figure 3.9: warm color



always. So except contrast computing, we will add this warm color element inside. The expression is like this:

$$WarmcolorMap = \begin{cases} 1 & if\ hueangle\epsilon[315, 360] \\ 0 & else \end{cases},$$

And effect of the Warmcolor Map will be expressed as Figure 3.9.

### 3.1.4 Color Map generation

So far, we have already computed the values of the Hue, Saturation, Intensity and Warmcolor Map for the Color Map, and all of them have been normalized to the value scaled 0 to 1, right now, the job is to combine them to generate the Color Map. Here we have two methods to do that:

1. Just as a summing form:

   $$ColorMap = Normalized(HueMap \times k_1 + SaturationMap \times k_2 + IntensityMap \times k_3 + WarmcolorMap \times k_4),$$

   the normalizing way is just as we mentioned above, and according to HVS, hue plays the most important part of all, saturation and intensity will be the next, last is warm color, so we define $k_1 = 5,\ k_2 = k_3 = 2,\ k_4 = 1$. The Figure 3.10 shows the result of this way.

2. We will compare the values of each position in the Hue, Saturation and Intensity Map, and then choose the greatest one to keep, then we take the Warmcolor Map as weight tool, if the position is warm color section, we will increase 25% of the value, if not, keep it still, and then do the normalization as before to get the final IM (example shown in Figure 3.11).

Figure 3.10: way 1 of Color Map

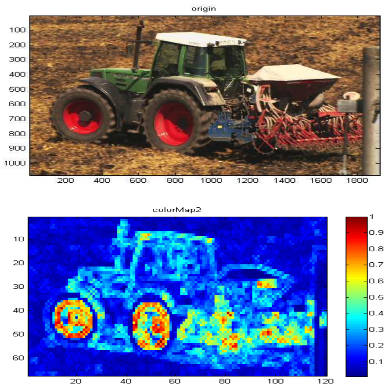


Figure 3.11: way 2 of Color Map

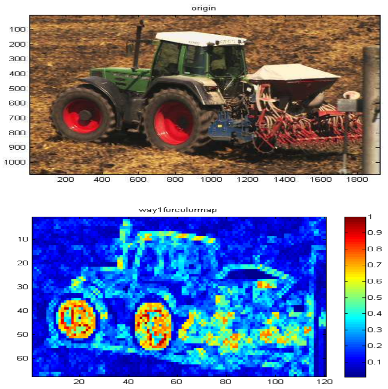
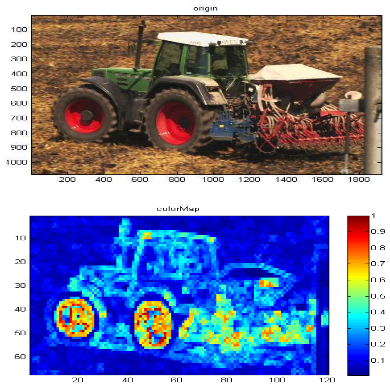


Figure 3.12: Final Color Map



If we compare these 2 ways, we will find out although the sensitive place has a higher value in way 2, its background's value seems too high for the Color Map, so we decide combine these 2 methods with a threshold. The threshold will work like this: if in the way 2 Color Map, the value is bigger than it, then we will keep it to the final Color Map; if not, we will change this value as the same place value in the way 1 Color Map, and use this value to the final Color Map, the expression is like

$$\begin{cases} if\ ColorMapWay2_n > threshold & ColorMap_n = ColorMapWay2_n \\ if\ ColorMapWay2_n <= threshold & ColorMap_n = ColorMapWay1_n \end{cases},$$

We from the examples, find out smaller than 0.4 Color Map value, the way 1 has a good value, and bigger than 0.4, way 2 is more reasonable, so we define the threshold is 0.4. And according this, we can combine the 2 ways, Figure 3.12 shows the result, this is our way to get the Color Map.

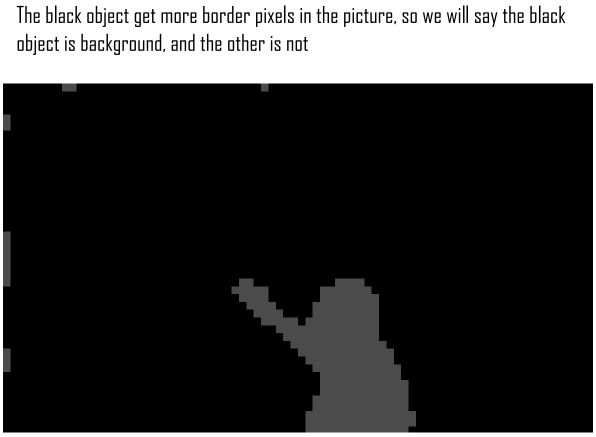
## 3.2 Size Map

In the computing, if pixel number of a object is greater than another, then we can say that this object size is bigger than another, in our working, we have already finish the object's segmentation, so there is not any problem we can get the pixel number of every object, and it is very helpful for the Size Map calculation. In our method, we will consider two conditions, and before that, we have to find out which object is the background.

### 3.2.1 Background

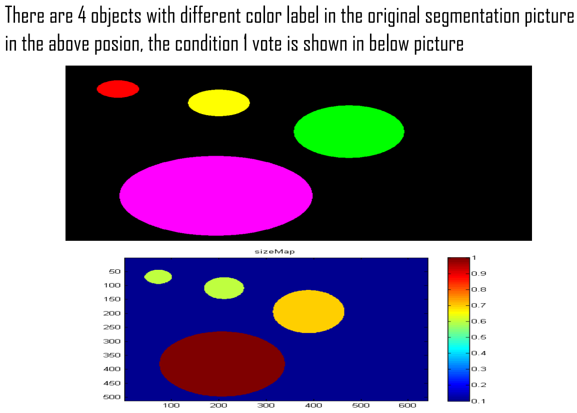
There are several ways to find out which object is the background, and here we will use this algorithm: we will compare the border pixel number of each object, and then treat

Figure 3.13: Background



The black object get more border pixels in the picture, so we will say the black object is background, and the other is not

Figure 3.14: The bigger object get the bigger vote value



There are 4 objects with different color label in the original segmentation picture in the above posion, the condition I vote is shown in below picture

the object which has greatest border pixel number as the background (Figure 3.13). Because usually background is not so important in the video or image as the objects in attracting the people's vision, so once we find out which object is background, we will firstly give it a smaller value as Background Map.

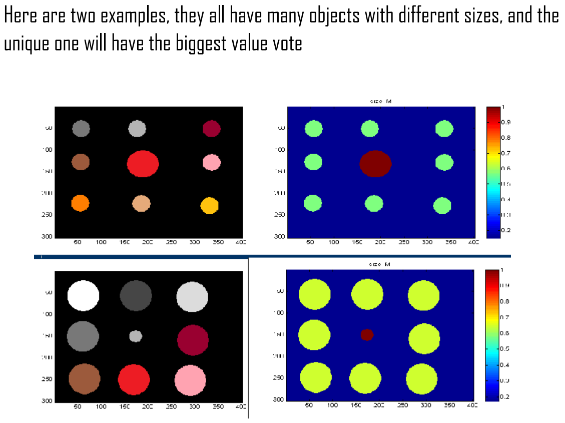
### 3.2.2 Size Map conditions

Here we have two conditions to generate the Size Map:

1. As the Figure 3.14 shows, except the background, the object which get the bigger size which will take a bigger vote to Size Map.

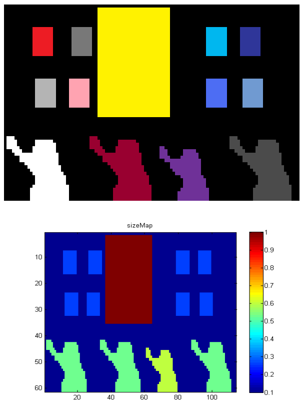
2. As the paper of Zaheer Aziz and Mertsching [3] said, the only obvious feature to determine salience is the size of objects and the uniquely sized objects are the obvious attractors of attention, it shows in Figure 2.5, so we will give the unique size object high contribution value (Figure 3.15).

Figure 3.15: The unique object has a higher value



Here are two examples, they all have many objects with different sizes, and the unique one will have the biggest value vote

Figure 3.16: Complex size example

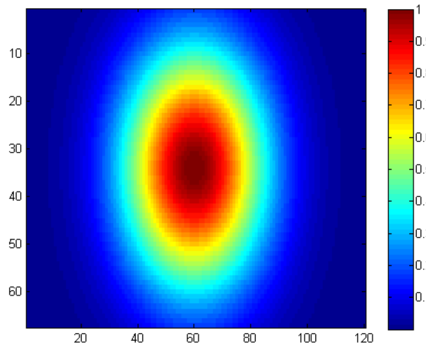


And according the 2 conditions of the above, we can get the Size Map, here is an example with the complex segmentation picture shown in Figure 3.16.

## 3.3 Position Map

In a frame, if one object is in an obvious position, it will attract people's looking, in Osberger's research [9, 11, 10, 8], the center of a picture is the most important position, because people used to focus on the picture's center. So we can say that, the center position of an image has a highest Position Map vote, and as it go to the place further and further away from the center, the vote will be less and less. Here we will use the Gaussian distribution on it, we treat the center point of the image as the peak of the

Figure 3.17: Position Map



Gaussian distribution, and then it will expand as the formula:

$$PositionMap = \frac{1}{2 \times \pi \times sigma^2} \times exp(-\frac{(x - \frac{width}{2})^2 + (y - \frac{length}{2})^2}{2 \times sigma^2}),$$

(shown in Figure 3.17). In the expression, the width and length are the resolution width and length of the image, and the x and y are the pixel point axis values in the image.

## 3.4 Motion Map

As the 3 Map we discuss above, they are all for image, our IM is for the video, for the moving objects, so we still need one other feature that can indicate the motion, here is the Motion Map coming out. Thanks for the segmentation, we know every object's situation, include the motion vector, from the different values of the motion vectors, we will get the different Motion Map value, and our method is like this:

1. First of all, you should find out the background objects, we have done that before in the Size Map calculation, and then we can get the background motion vector;

2. Then use motion vector of the back ground as a compensation value, and use every object's to subtract it to get a value after compensation (the background motion will become 0 at that time), it can show the real movement in the frame;

3. Use the follow rule to compute the Motion Map:

$$NormMV = \sqrt{M_x^2 + M_y^2},$$

Figure 3.18: video frame and its segment result

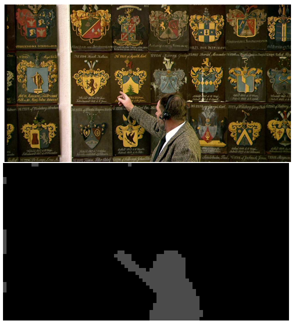
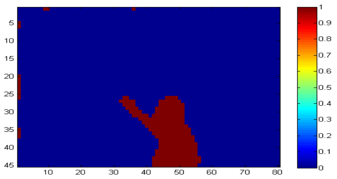


Figure 3.19: The Motion Map of the example



$$MotionMap_{object(n)} = \begin{cases} if\ NormMV_{object(n)} \epsilon [0,7), & \frac{NormMV}{7}; \\ if\ NormMV_{object(n)} \epsilon [7,36), & 1; \\ if\ NormMV_{object(n)} \epsilon [36,96) & -\frac{NormMV}{60} + \frac{8}{5}; \\ else & 0. \end{cases},$$

where $M_x\ and\ M_y$ is the value of the motion vector after the compensation.

After that, we can generate the Motion Map for a video frame, here is an example:

Figure 3.18 shows a video frame and its segmentation result, and different color label values express different objects.

And its motion vector is shown in here:

$$\begin{cases} label = 8421376; & px = 1, py = 0 \\ label = 8421456; & px = 0, py = 0 \end{cases},$$

where the $px\ and\ py$ means the motion value in the x and y axis, then use our method, we can get the Motion Map shown in Figure 3.18.

# 4 IM generation

By finishing computing the Motion, Position, Color, Size Map, we need to combine them to get a IM, in this part, we will discuss the way to combine these elements value to get the good IM. Figure 4.1 gives us 3 videos, they are “shields”, “tractor” and “calendar”: “shields” is describing the course of a guide walking and pointing the shields; “tractor” is expressing a tractor is moving in different position; and “calendar” is showing us a scene that a calendar and a moving toy train. Here we will try to use 4 ways to do them, and check the results to make the decision, these 4 ways are the most usual used by people.

## 4.1 Combination ways

### 4.1.1 Straight sum way in same rate

Way 1 is just sum the value of the 4 maps straightly, and normalized it at last to get the IM, its expression:

$$IM = normalize(MotionMap + SizeMap + ColorMap + PositionMap),$$

and Figure 4.2 has shown us the results frame IM in this way.

Way 1 get a good IM results, but for the “shields” frame, it seems that the center background’s votes is not enough.

Figure 4.1: Example video frames

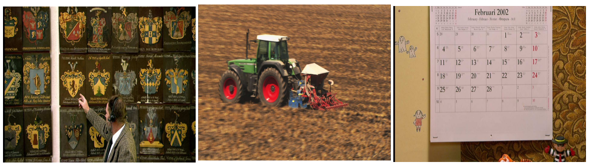


Figure 4.2: Way 1 results

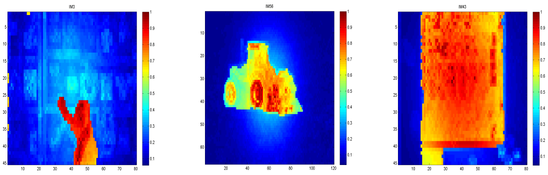
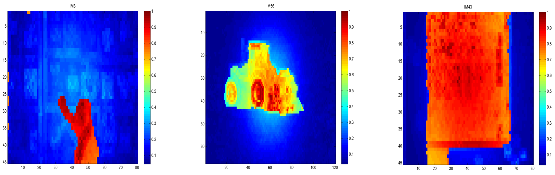


Figure 4.3: Way 2 results



### 4.1.2 The sum with different rate

It is for the video, so Motion Map should take more votes than the other maps, we still take their sum and normalized it, but we will give more parts from Motion Map, the expression is:

$$IM = normalize(k \times MotionMap + SizeMap + ColorMap + PositonMap),\ k > 1,$$

we will take k=2 in here, and here are the results in Figure 4.3.

The result seems not bad also, but the “shields” frame background still not good, and the thing in down left position in the “calendar” frame is decreased because of the increase of the objects movement’s influence.

### 4.1.3 The sum of Spacial and Motion multiply Position

In way 3, we firstly get the Spacial Map of the frame, which comes from the summing the Size Map and Color Map and then normalize it; then we combine the Spacial Map with Temporal Map which is Motion Map in summing and normalization; last we use the Map we get be multiplied by the Position Map to get the IM, the processing is like this:

$$IM = normalize(PositionMap \times Map_{SpatialandTemporal});$$

Figure 4.4: Way 3 result

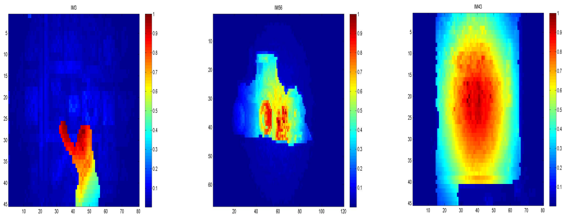
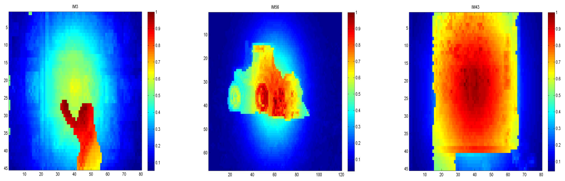


Figure 4.5: Way 4 result



$$SpatialMap = normalize(SizeMap + ColorMap);$$

$$TemporalMap = MotionMap;$$

$$Map_{SpacialandTemporal} = normalize(SpatialMap + TemporalMap).$$

And here is the results of this way (Figure 4.4):

### 4.1.4 The sum of Spacial, Motion and Position

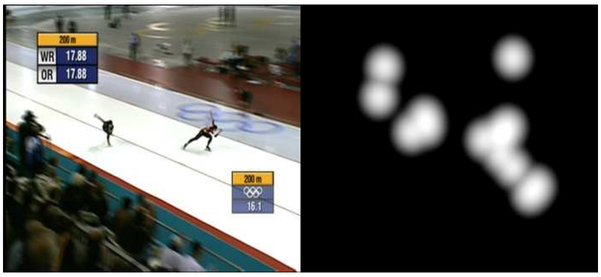
Way 4 is nearly like the way 3, way 3 last processing uses the Position Map to do the multiplying, and we will change it to the adding, the expression is shown in this way:

$$IM = normalize(PositionMap + Map_{SpatialandTemporal}).$$

Figure 4.5 show this way's result:

We find out that in way 4, "shields" background is get a sensible value, but it seems still too concentrated in center part and too neglect the surround views.

Figure 4.6: An example of the eye tracking experiment result



## 4.2 Compare with the eye tracking experiment on the sequence

Eye tracking experiment is a way of exploring what people look at in any given situation and record their visual attention strategies with the location and duration of their fixations. Just as the Figure 4.6 show, the white points show us in the screen the point people focus on.

The situation of the eye tracking experiment results is come from the all the frames, every frame has its own eye tracking experiment frame, however this is not match with the our IM calculation condition, there are 2 points:

1. In our IM result, every block has its own value, the value scale from 0 to 1, but in eye tracking experiment, the important position pixels will get the value, but the others will not have.
2. In our IM method, we are using GOF as a unit to compute, it contain all the frames features of the GOF, but in eye tracking experiment, the unit is a frame.
3. In our IM algorithm, we are facing HDTV high resolution video, but in this eye tracking experiments, the video sequences are all much lower than ours.

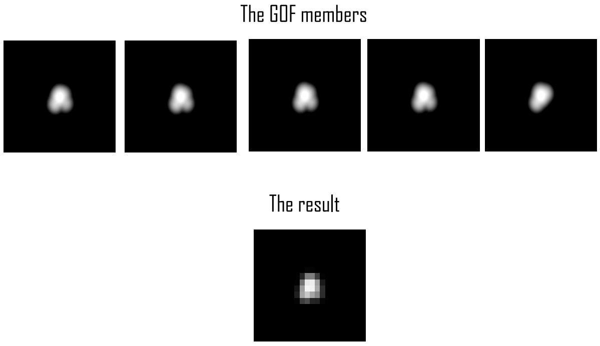
So we should do some change on the eye experiment result, first of all, we change the value on pixels to value on macro block, we will use this expression:

$$value_{macroblock} = \frac{\sum_{i=1}^{the\ macroblock\ pixels} value_{pixels}}{the\ macroblock\ pixels}.$$

Then we change the unit of generating the eye tracking experiment result to GOF, here is the method:

$$value_{GOF} = \frac{\sum_{i=1}^{GOFnumber} value_{macroblock}}{GOF\ number}$$

Figure 4.7: transform the value of pixels to value of macro blocks



so the Figure 4.7 shows an example of the result:

Even so, the difference of the eye tracking experiment and IM calculation is still very large, because the former just contain the most important places and the least important places, and our method will classify the place as different value of their importance, not just two kind value, these two experiments are hardly totally fit. However, we can calculate the correlation coefficient of the eye tracking experiment with the 4 different ways, it at least can show which is better one in these 4 ways. We will use below 3 sequence Kayak, PatinVitess and Canoe to do it (Figure 4.8, Figure 4.9 and Figure 4.10), then we compute the correlation coefficients of them to do the comparison, here is the expression:

$$cc = \frac{\sum_{i=1}^{macroblock\ number}[(x_i - \overline{x})(y_i - \overline{y})]}{\sqrt{\sum_{i=1}^{macroblock\ number}(x_i - \overline{x})} \times \sqrt{\sum_{i=1}^{macroblock\ number}(y_i - \overline{y})}},$$

where $x_i$ *and* $y_i$ are the macro block values of the IM and macro block values of the eye tracking experiment in a frame, $\overline{x}$ *and* $\overline{y}$ are the average value of all the blocks in a frame in IM way and eye tracking experiment way.

Figure 4.11 (Kayak), Figure 4.12 (PatinVitess) and Figure 4.13 (Canoe) show the features in axis.

As these figures show, we hardly to say which one is the best. And some of the points get the bad value, we have to say, first of all, may be the segmentation is not fit for the sequence (we use the HDTV sequence tool), the IM result is depending on the segmentation. Then we calculate the average value for the comparison and put them in the Table 4.1:

Figure 4.8: Kayak

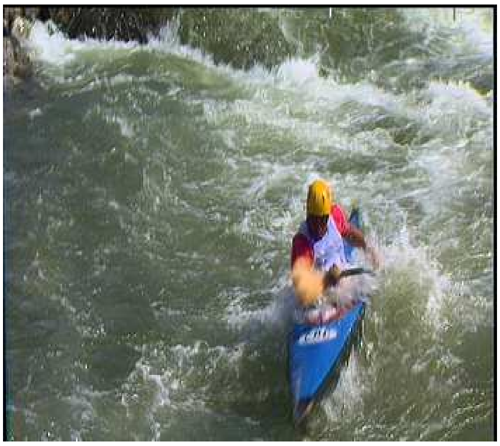


Figure 4.9: PatinVitess

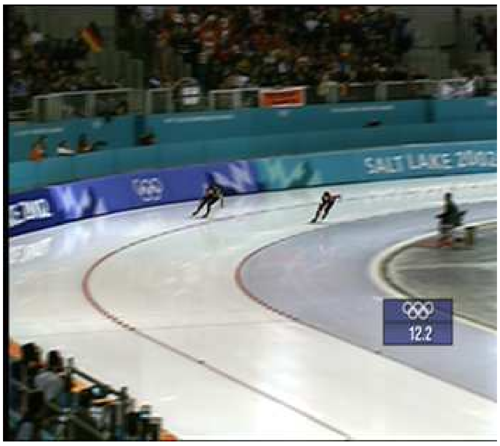


Figure 4.10: Canoe

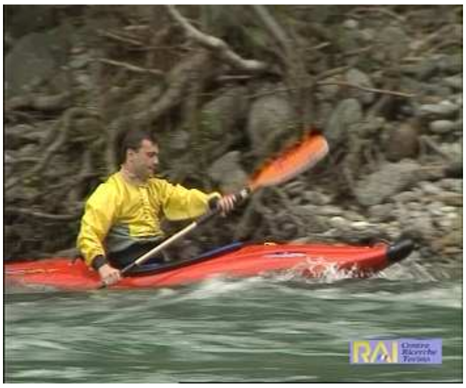


Figure 4.11: Correlation coefficient of the Kayak (x-axis is for the sequence frames, y-axis is for the cc value)

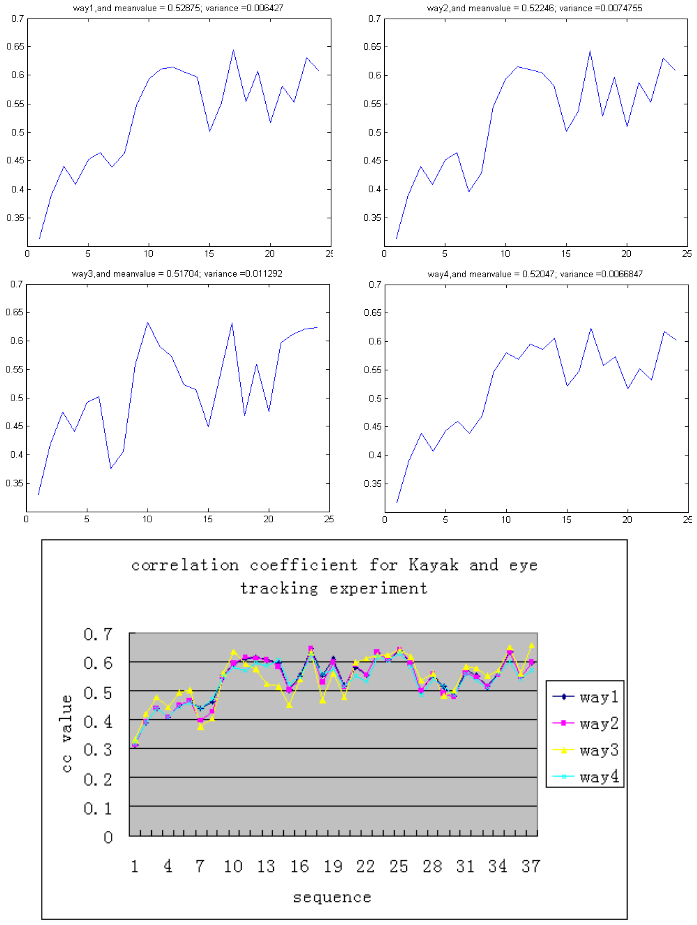


Table 4.1: Average cc values for the 3 sequence

| The ways | way1 | way2 | way3 | way4 |
|---|---|---|---|---|
| mean cc for PatinVitess | 0.586 | 0.589 | 0.602 | 0.576 |
| mean cc for Kayak | 0.539 | 0.533 | 0.537 | 0.529 |
| mean cc for Canoe | 0.483 | 0.464 | 0.432 | 0.493 |
| mean cc value of the 3 sequence cc values | 0.536 | 0.529 | 0.523 | 0.532 |

Figure 4.12: Correlation coefficient of the PatinVitess (x-axis is for the sequence frames, y-axis is for the cc value)

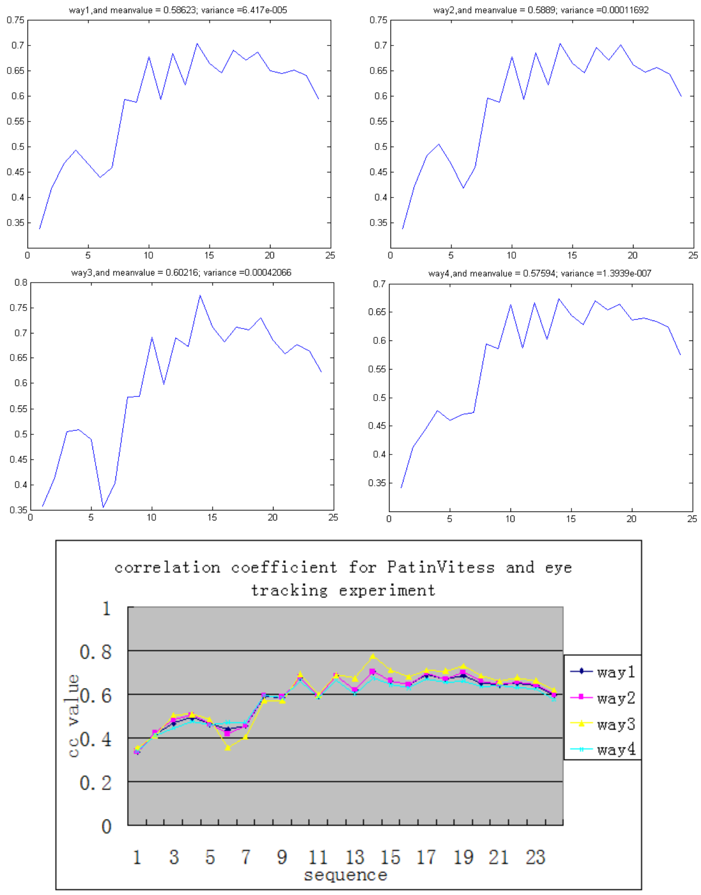


Figure 4.13: Correlation coefficient of the Canoe (x-axis is for the sequence frames, y-axis is for the cc value)

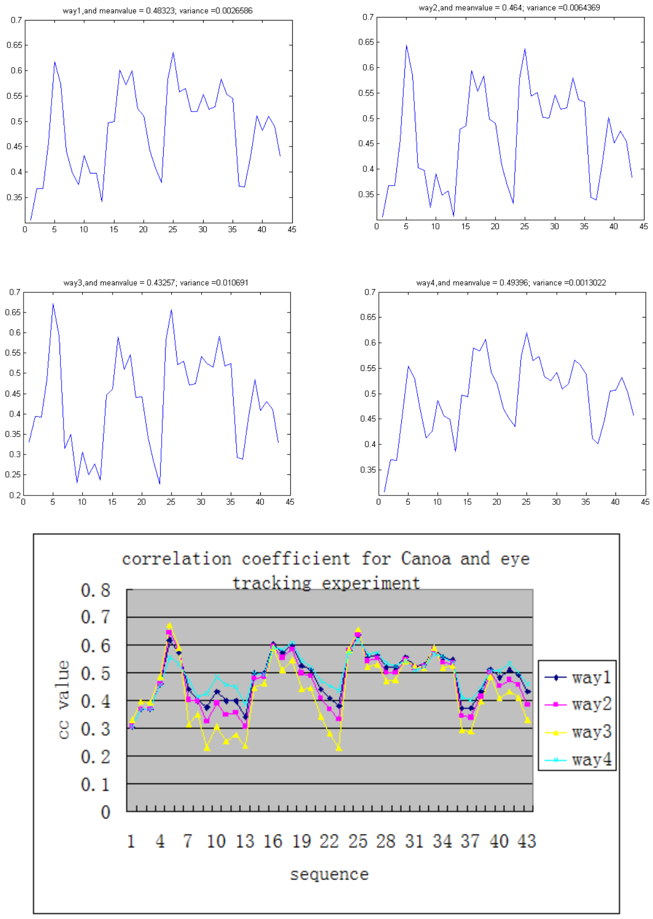
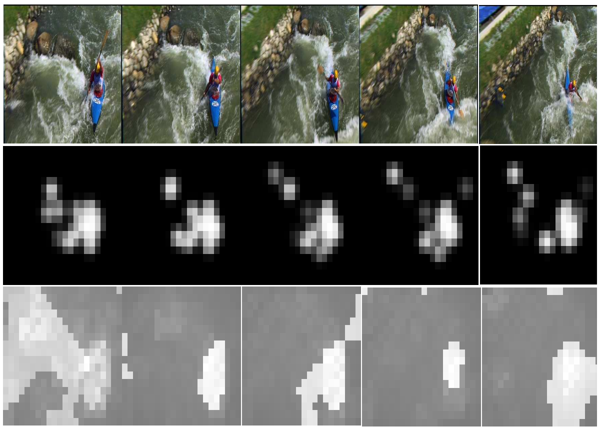


Figure 4.14: comparison sequence of Kayak



As the mean values show, for the 3 sequence the way 1 is the best way of all. Here is an example sequence comparison in PatinVitess and eye tracking experiments in Figure 4.14.

# 5 Conclusion

In this report, we have done some Important Map research after the segmentation finished. According to studying from the HVS and papers of other researchers, here we discuss a new way for IM calculation, this project is work for "Contribution to the conception of a analysis tool of the video before its coding using H.264/AVC", and we use the size, motion, position and the color from the frame to deduce an IM calculation tool, this tool here is capable of accelerating the quantization step decisions (choice of the Qp) and to adapt the coding strategy in order to improve the visual quality of the decoded video. We believe that the final algorithm will be implemented in a massively parallel architecture for real-time coding.

# Bibliography

[1] *About IRCCYN.* http://www.irccyn.ec-nantes.fr/.

[2] *H.264 and MEPG-4 VIDEO COMPRESSION Video Coding for Next-generation Multimedia.* WILEY Editorial Offices, 2003.

[3] Muhammad Zaheer Aziz and Barbel Mertsching. Fast and robust generation of feature maps for region-based visual attention. *IEEE TRANSACTIONS ON IMAGE PROCESSING, May 2008, Volume: 17, Issue: 5 On page(s): 633-644 ISSN: 1057-7149*, 2008.

[4] Olivier Brouard, Fabrice Delannay, Vincent Ricordel, and Dominique Barba. Robust motion segmentation for high definition video sequences using a fast multi-resolution motion estimation based on spatio-temporal tubes. *Picture Coding Symposium, November,Portugal*, 2007.

[5] Olivier Brouard, Vincent Ricordel, Fabrice Delannay, and Dominique Barba. Spatio-temporal segmentation and regions tracking using a markov random field model. *ICIP, San Diego, USA*, October 2008.

[6] D. Chai and A. Bouzerdoum. Coding videophone sequences at better perceptual quality by using face localization and bit redistribution. *IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)Volume 9, Issue 4, Jun 1999 Page(s):551 - 564 Digital Object Identifier 10.1109/76.767122*, 2000.

[7] Olivier Le Meur, Patrick Le Callet, Dominique Barba, and Dominique Thoreau. A conherent computational approach to model bottom-up visual attention. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, May, 2006, pp.802-817.

[8] Wilfried Osberger, Sean Hammond, and Neil Bergmann. An mpeg encoder incorporating perceptually based quantisation. *IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunications ISSUE, 2-4 Dec, Australia*, 1997.

[9] Wilfried Osberger and Anthony J.Maeder. Auotmatic identification of perceptually important regions in an image using a model of the human visual system. *14th International Conference on Pattern Recognition, August, Australia*, 1998.

[10] Wilfried Osberger, Anthony J.Maeder, and Neil Bergmann. A perceptually based quantization technique for mpeg encoding. *SPIE Human Vision and Electronic Image III 3299, USA, January*, 1998.

[11] Wilfried Osberger, Anthony J.Maeder, and Neil Bergmann. A technique for image quality assessment based on a human visual system model. *EUSIPCO September, PP 1049-1052, Greece*, 1998.

[12] Chih-Wei Tang, Ching-Ho Chen, Ya-Hui Yu, and Chun-Jen Tsai. A novel visual distortion sensitivity analysis for video encoder bit allocation. *2004. ICIP '04. 2004 International Conference on Image Processing, Publication Date: 24-27 Oct. 2004, Singapore*, 2004.