Performance enhancement of animated images compression using videoconferencing Techniques

Shima'a kthir saleh
Education College, Al-Mustansiriyah University

Abstract

A digital image is defined as visual representation of meaningful data within spatial domains. Aarranging certain number of digital images successively is called animated image or video. Digital video is video information that is stored and transmitted in a digital form. Digital video has been used for a number of year ,for example, in the television broadcasting industry. Most digital image contain high degree of redundancy, which means that an efficient compression technique can significantly reduce the amount of information needed to store or transmit them.

Special techniques work to reduce the degree of information redundancy, which can be found between single pixels, between lines, or between frames, when ascene is stationary or slighty moving. In this paper,H.263 technique has been adopted. The structure of this technique and characteristics has been studied. Moreover, factors that limit the efficiency of this technique has been identified throught implementing and analyzing the well_known motion search techniques
2D_Logarthmic search, conjugate search (one time search) . In this paper, two new techniques have been developed, threshold quarter_pixel and coddeven motion search to enhance H.263 technique efficiency. The first technique increases the image smoothness of video frames. The second decreases the elapsed time of video compression(encoding delay time)

1-Introdduction

In the this paper, four well_known block matching search techniques were reviewed as well as the strategy used in each of them.These techniques differ from each other. Each one has advantages and drawbacks affecting directly the effectiveness of H.263 CODEC.

In this paper, two considerably improved techniques are developed to enhance the efficiency of H.263 technique. This is done through enhancing video image quality and reducing encoding delay time limitations.

The first technique (called thresholding qurater_pixel technique) greatly intensifies the performance of the traditional quarter-pixel technique by adding threshold value controls clarity and smoothness for each of the pixel groups included from which video frames are composed.

The second technique (called oddeven search technique) employs a new search strategy to match similar blocks from video successive frames, and exclude the similar blocks. This technique achieves an efficient compression process. These techniques are explained below.

2-Thresholding quarter-pixel Accuracy Improvemen

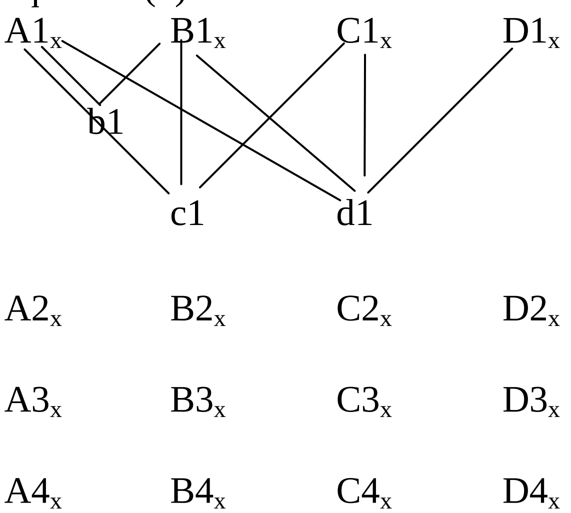
As we mentioned earlier, H.263 uses sub_pixel accuracy for motion compensation instead of using a loop filter to smooth the anchor (refernce) frame as in H.261. the quarter _pixel technique achieves the image smoothing by considering a pixel and its neighbors, eliminating any extreme values (high frequency_noise) in this group.

Quarter-pixel technique dose not take into account the color distance relationship among pixels directly, but rather it produes values representing quarter the distance among these pixels.

In this paper, we improved the technique produces by adding an equilibrium factor to achieve high level of smoothness among anchor frame pixels through computing threshold value.

Threshold value stands for the average difference between among of pixels which reduce the noisy pixels values (not correlated pixels) resulted from quarter_pixel technique. This makes pixels values of each group closer, hence leading to enhance the video anchor frames.

The improvement of quarter_pixel accuray is explaining in Figue(1) and equation (1)?



Where x integer_pixel postion

a1=A1

b1=((A1+B1)/2)-Thr

c1=((A1+B1+C1)/3)-Thr

d1=((A1+B1+C1+D1)/4)-Thr

a2=(A1+B1+C1+D1+A2/5)-Thr
b2=(A+B+C+D+A2+B2)/6)- Thr
c2=(A+B+C+D+A2+B2+C2)/7- Thr
d2=(A+B+C+D+A2+B2+C2+D2)/8)- Thr

.
.
.
.

d4=(A1+B1+C1+D1+A2+B2+C2+D2+A3+B3+C3+A4+B4+C4+D4)/16

Figure(1):threshold quarter-pixel method

Where A, B, C, and D are grouped pixels and *Thr is* an additional value used to reduce the extreme quarter_pixel values. According to equation (1),threshold value will decrease color distance for group(a, b ,c, and d) and produces a new group of pixels with suitable color approximation.

3-Oddeven search technique(OES)

The main idea of the proposed oddeven(OES) search algorithm starts by computing the cost function(Typically, MAD function is used) for the odd blocks of x-axis and for the even blocks of y-axis finding the best match .

The matching candidate block contains the minimum value of the computed cost function values in the search windows of the previous frame. This block represents the most similar for the selected block in the current frame of the video sequence.

The matching process starts by calculating the cost function of the odd locations of x-axis and even locations of y-axis in the search window. The location that produces the smallest cost function becomes the center location for the next step, and the search range is reduced by half.

The matching process of the OES algorithm will continue unitl achieving the condition that the minimum computed cost function value is equal or less than threshold value previously determined to achieve the matching process. Threshold value is determined, depending on the requirements of the applications that use the proposed algorithm. The threshold value controls the speed and efficiency of the block matching process. This effectively affects the H.263 bit rate and encoding delay time.

Figure (2) shows the block matching process on the y-axis. Figure(3) shows the block matching process on the x-axis

3-1-Oddeven Control Paramters

This technique uses two control values to produce the best and fastest block matching process. These control values are:

.Information Density Control Value: A value is determined representing number of bits from which the maximum color value of the matched block is composed. This value represents the amount of color information distributed of matched block. The following formula is used to compute this value:

MAD*Threshold=[Log2(Max(SBLK)*)]……(2)

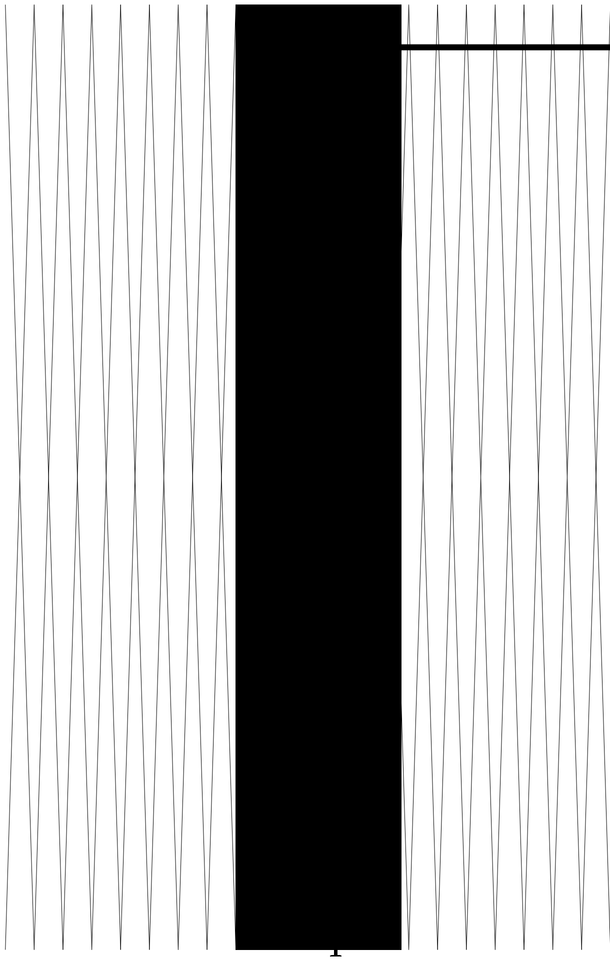
Where, SBLK represent the current frame candidate block for matching process. The computed values. Of MAD threshold is compared with the minimum MAD computed value If the selected MAD minimum value is less or equal to MAD Threshold value , the matching process will be terminated. The block of the minimum MAD value is the approximated match of SBIK.

.Loop Step Size Control Value: Three value are useD (0.5,0.7 or 0.9) to reduce the matching process comparison on searching window blocks to find the best match. Where:

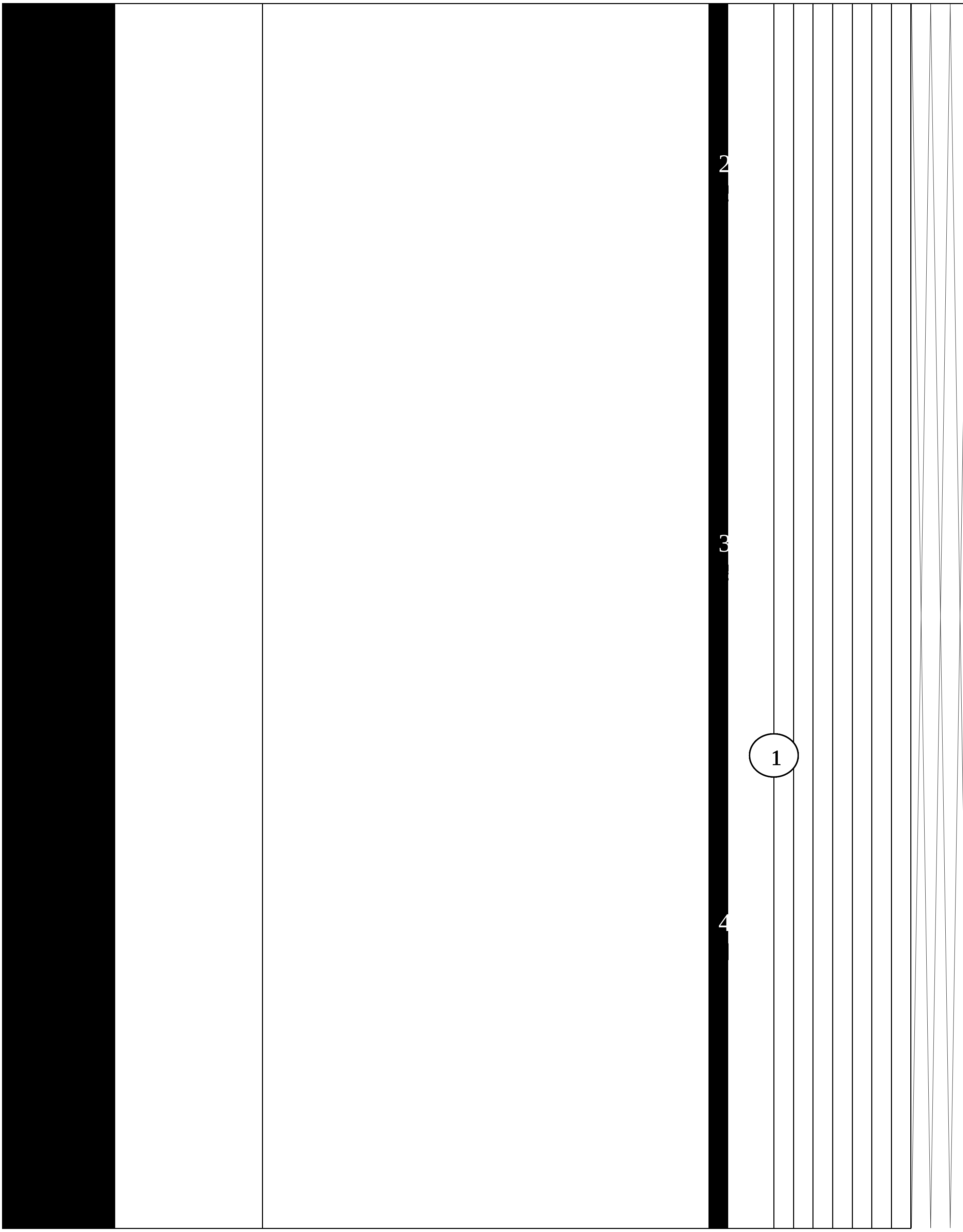
1- 0.5 value, divides the loop iteration by 2, this makes the OddEven technique compare half of searching window blocks.
2- With 0.7 value, OddEven technique compare one third of searching window blocks.
3- With 0.9 value, OddEven technique compare all of searching window blocks.

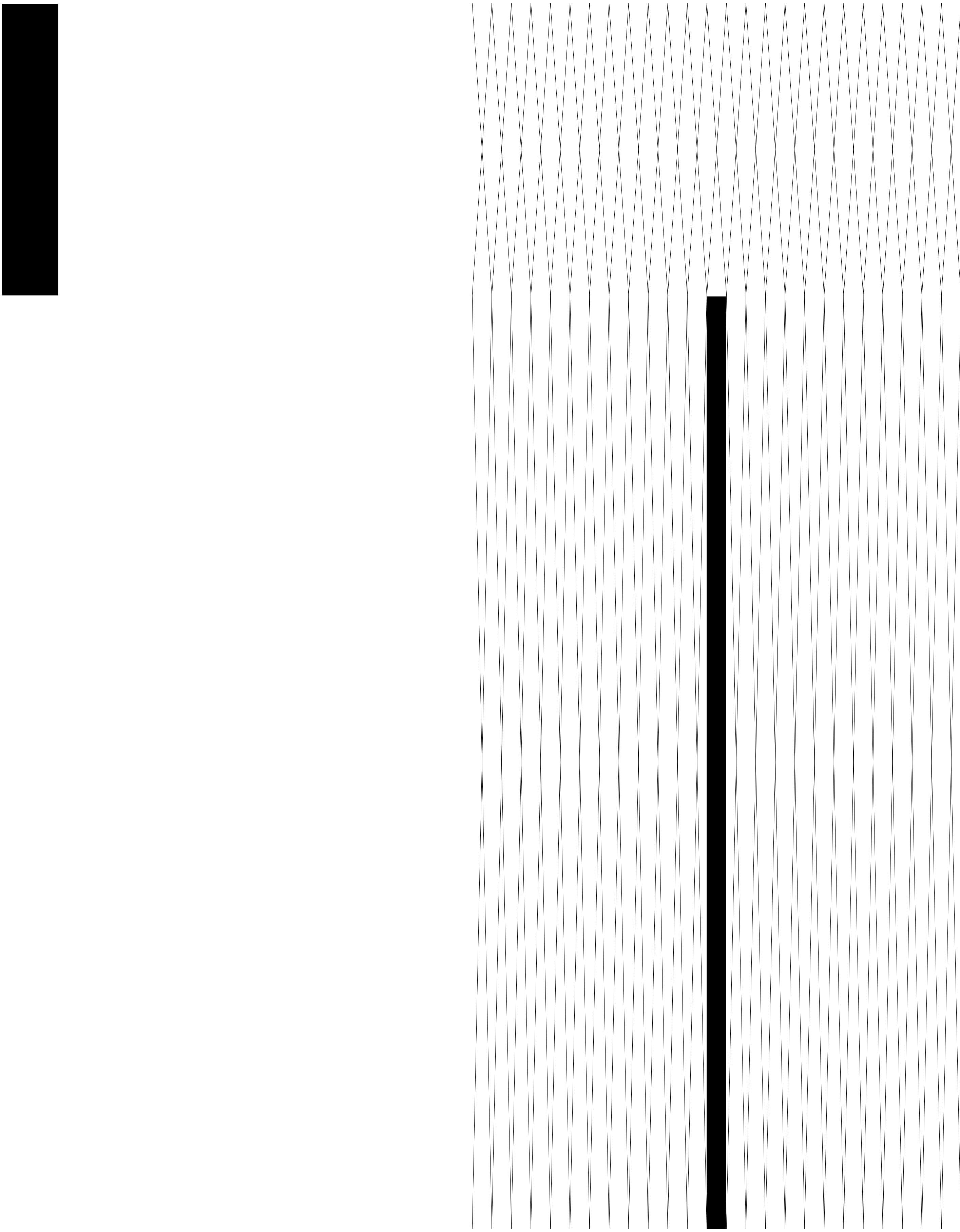
dy

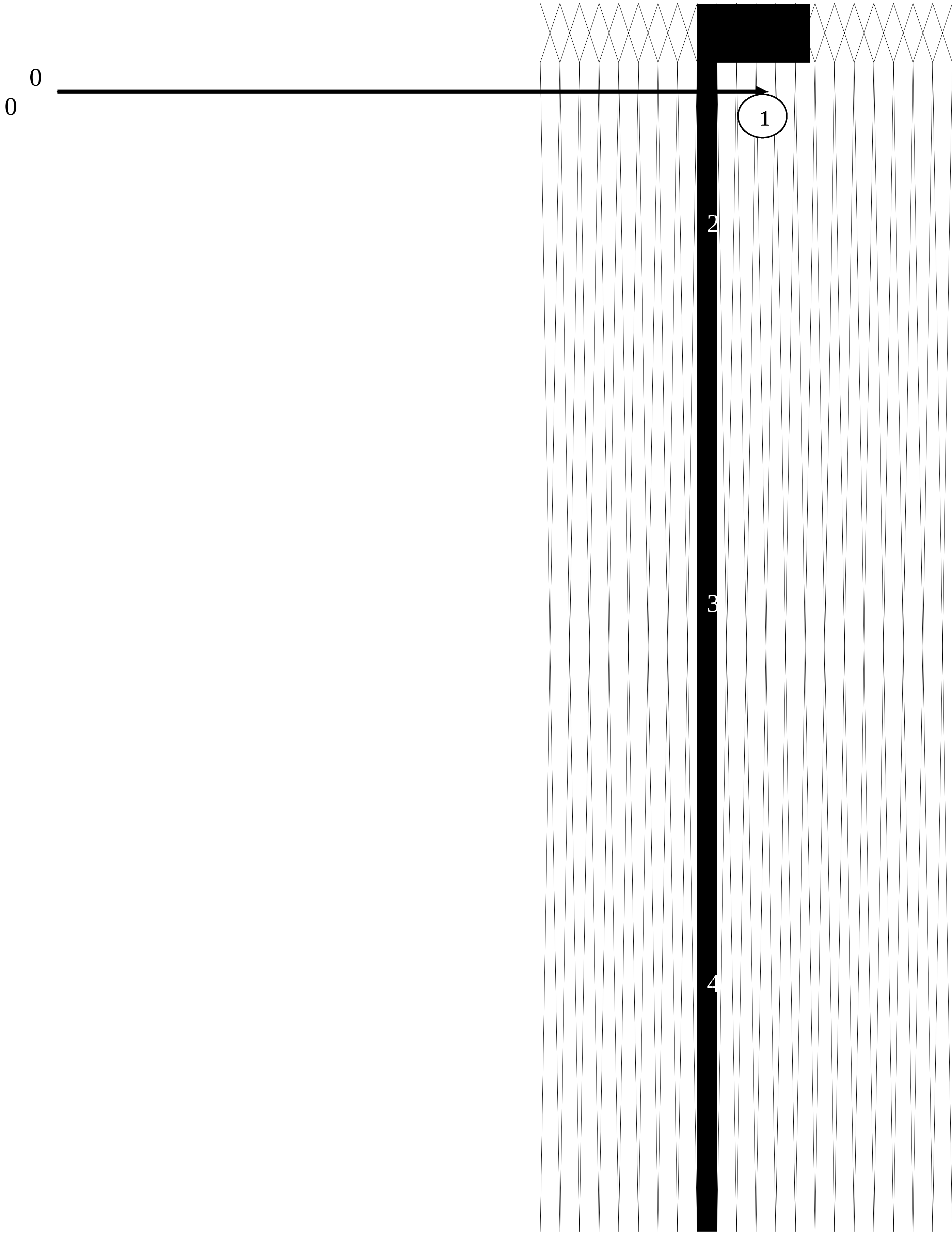
-
4

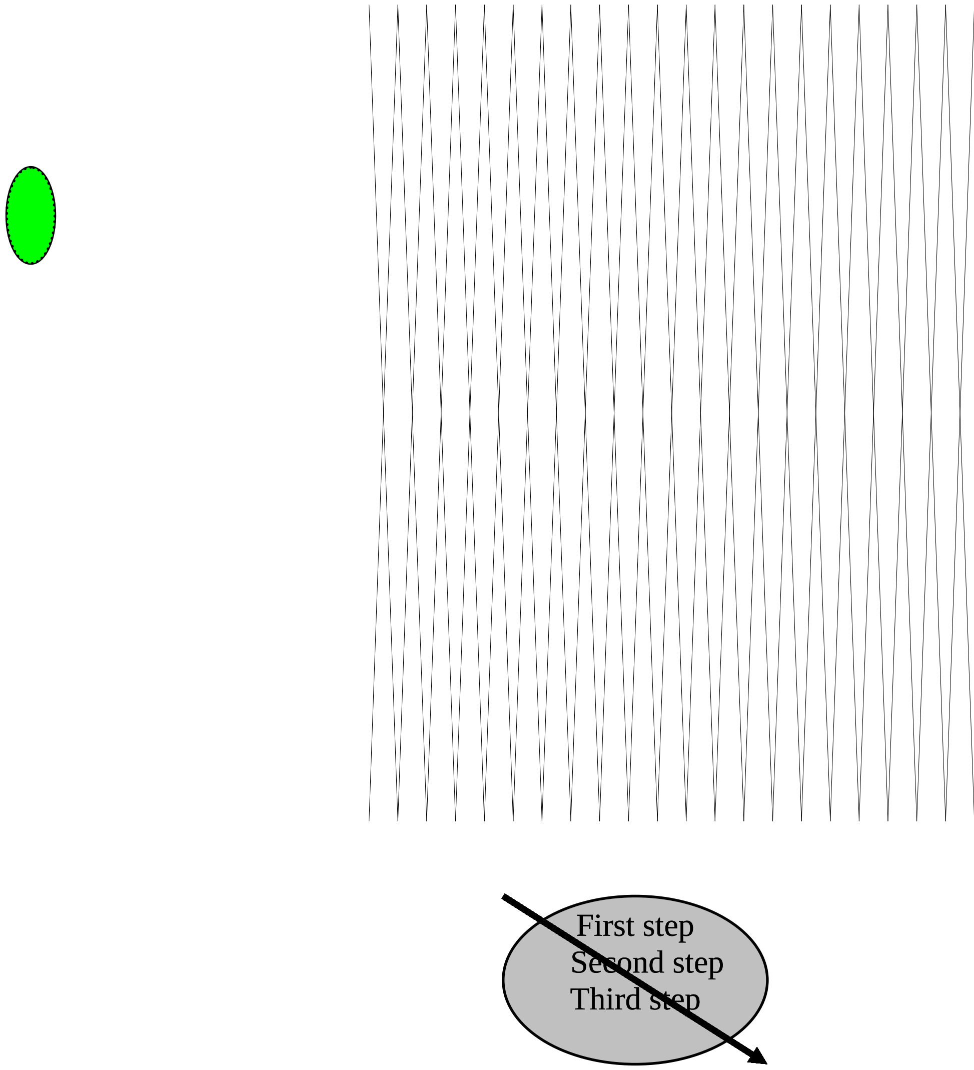


1

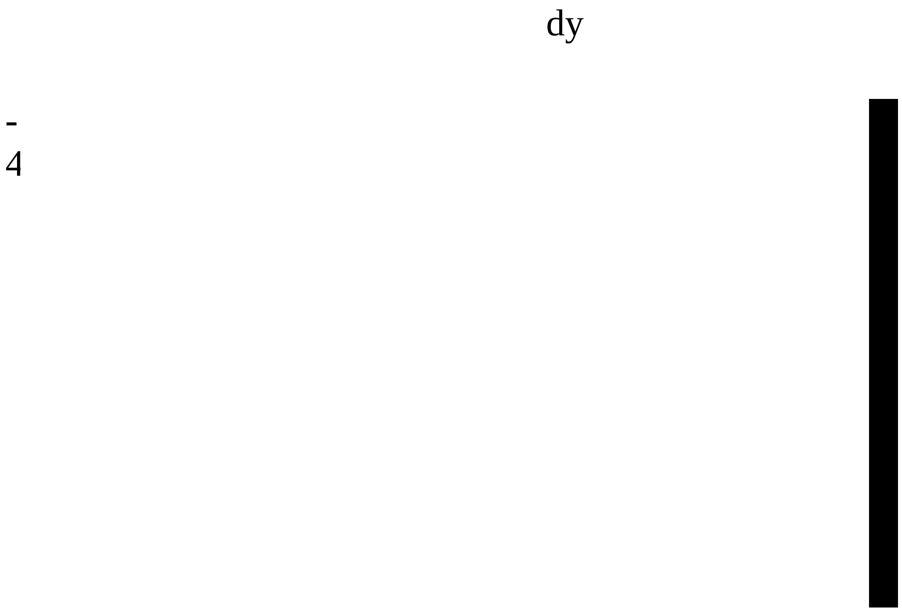

2
3
1
4

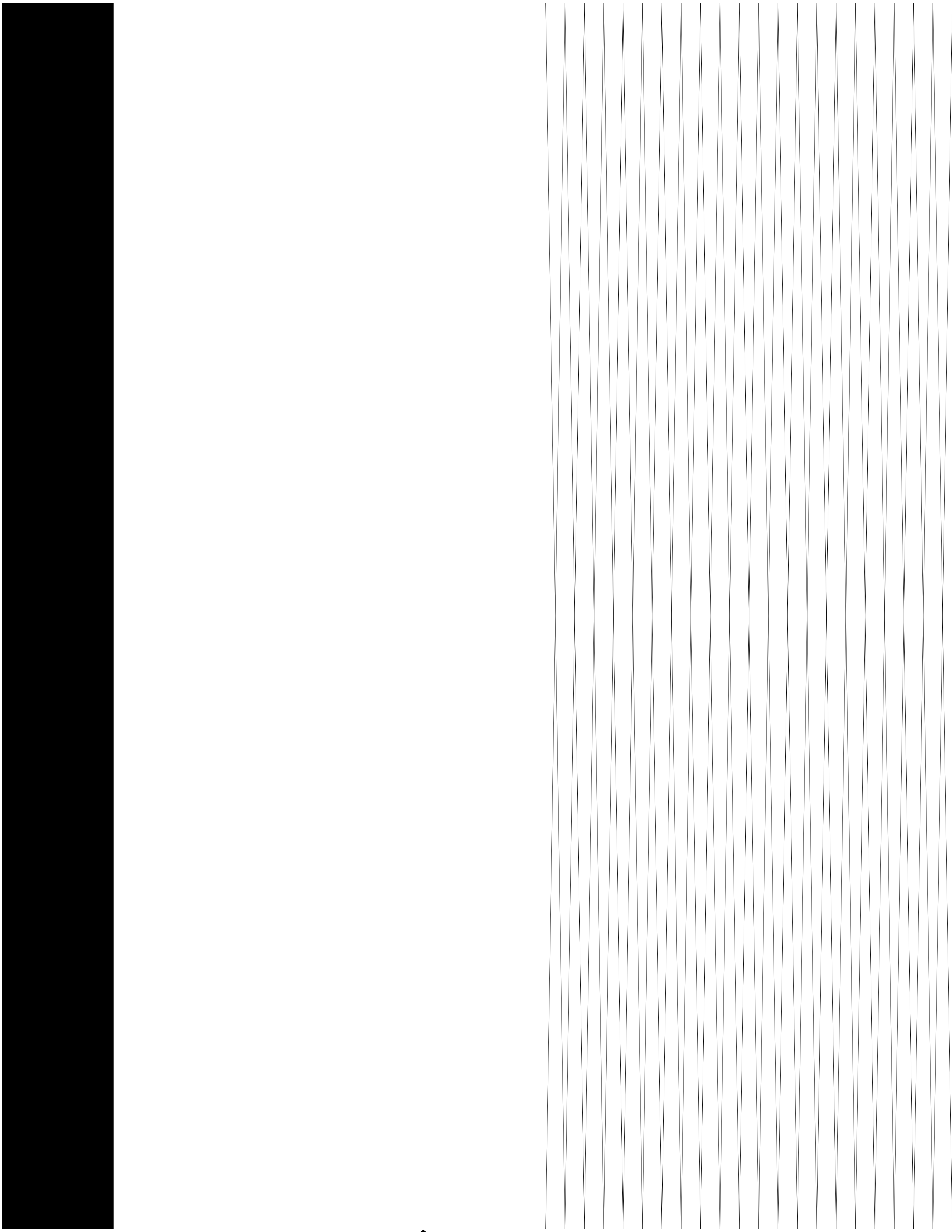


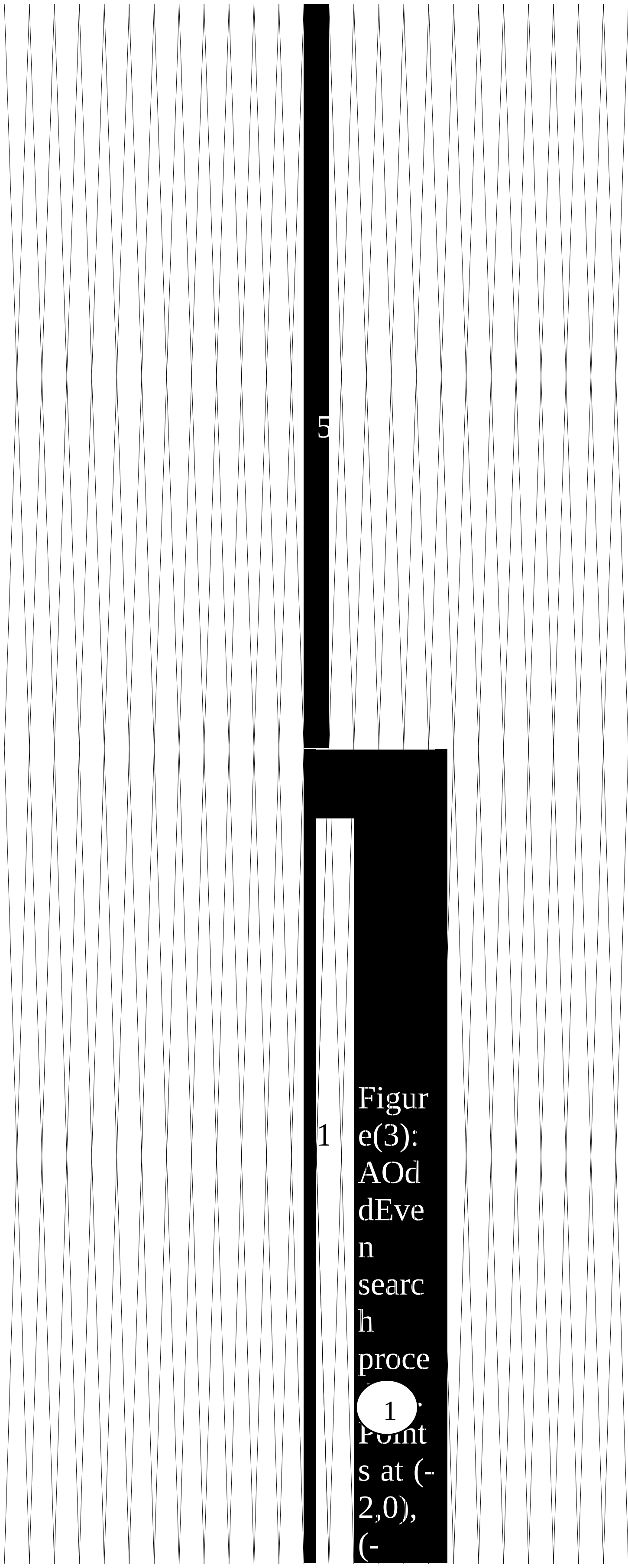

0
0
1
2
3
4



Figure(2):OddEven search procedure. Point at (0,-3),(-2,-3), and (-2,-2) are found to give the minimum dissimilarity in 1,2, and 3 respectively.







Figure(3): AOddEven search proce[illegible] Points at (-2,0), (-

2

imum dissimilarity in steps 1,2, and 3 res pectively.

4- OddEven Search Algorithm

INPUT

Swin

: The seaching window SBLK : Any block of th e current frame, $frame_i$

Stepsize : A value

used as a loop step size. This value takes

(0.5,0.7 or 0.9) Sr, Sc :Stands for width and height of search window (Swin).

OUTPUT:

M

V : Resulted motion vector prediction value

.

STEP 1: Compute the starting search location o f Swin, Ycnter=Sr div 2, Xcnter=Sc div 2,

[redacted]

Where, Y_{center} and X_{center} represent the cen ters of the search window. Initializing jum p variables, $Jump_i = size(\mathit{SBIK})*2$, $Jump_i = Jum$

p_i , Where , $Jump_i$ and $Jump_j$ represent Y-axis

[redacted]

and X-axis jump variables Initializing MAD variable, $Min_{MAD}=0$. Allocating fixed array size that holds the computed MAD values, Man_array=array of real.Compute an accurac y value used to control the search process

.

MAD

Threshold=[Log$_2$(Max($SBIK$))]Compute

t

he MAD of the center of Swin, Min_{MAD} = MAD(

Ycnter, Xcnter, Swin, SBIK), MADbits=[Log2(

Min MAD x 255)]

If MAD Threshold >= MAD bits Then

Extract motion vector ,Mv= position of min MA

D ,Go to step 5 ,Endif

STEP 2:

Initializing the MA

D counter: Let Count=1. Compute the MAD of t

[redacted]

he Y-axis odd blocks, For i=(Y cnter div 2

) to (Sc-JUMPi)*stepsize threshold step jum pi MAD-array(Count)=MAD(Ycnter,i, SWIN,SBL

K), Increment count by 1 EndFor

STEP 3:

Compute

the MAD of the X-axis even blocks.For j=(X center div 2) to (Sr-Jumpj)*StepSize thresho ld StepJump j MAD_array(Count)=MAD(j,Xcnter

, Swin , SBIK) . Increment Count by I , End for

STEP 4:

Find the minimum MAD value form MAD array(

MAD_array)

Min_{MAD} = Min(MAD-array). Comparin

g the Min MAD with MAD threshold and find t he motion vector.

$MAD\ bits = [Log_2(Min_{MAD} x 255$

)] If MAD Threshold >= MAD bits Then Extract mo tion vector, MV= position of Min_{MAD} , Go to

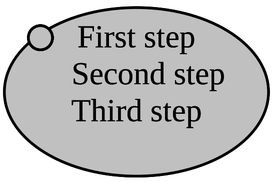
STEP5 ,Else Changing the location of Xcn ter and Ycnter according to MinMAN locatio n.

Y_{cnter}= The location Y of Min_{MAD}, X_{cnter}

= The location X of Min$_{MAD}$, Detecting the ne w values of the Xenter and Ycenter if they are equal or exceeded The search window boun daries ,If ((Xcnter+Jumpi)>=Sc)OR((Xcnter<=

1)) Then X_{center}= the Left or Right boundary value, Jumpi=jumpi div2, EndIF. If (Ycnter+j ump$_i$>=sr) or (Ycntr<=1) then Jump$_i$=jump$_i$ di v 2;Endif. Repeat the steps 2,3 and 4 until finding appropriate motion vector ,EndIf. **ST**

EP5: END.



Figure(3):AOddEven search procedure. Points at (-2,0), (-2,2), and (-3,2) are found to give the minimum dissimilarity in steps 1,2, and 3 respectively.

4-OddEven Search Algorithm

INPUT

Swin :The seaching window

SBLK :Any block of the current frame,frame$_i$

Stepsize :A value used as a loop step size. This value takes (0.5,0.7 or 0.9)

Sr, Sc :Stands for width and height of search window (Swin).

OUTPUT:

MV :Resulted motion vector prediction value.

STEP1:

Compute the starting search location of Swin,

$Y_{cnter}=S_r$ div 2,

X_{cnter}=Sc div 2,

Where, Y_{cnter} and X_{cnter} represent the centers of the search window.

Initializing jump variables,

$Jump_i$=size(*SBIK*)*2,

$Jump_j=Jump_i$,

Where,$Jump_i$ and $Jump_j$ represent Y-axis and X-axis jump variables

Initializing MAD variable,

$Min_{MAD}=0$.

Allocating fixed array size that holds the computed MAD values,

Man_array=array of real.

Compute an accuracy value used to control the search process.

MAD Threshold=[$Log_{2(}$Max(*SBIK*))]

Compute the MAD of the center of Swin,

Min_{MAD} =MAD(Y_{cnter},X_{cnter}, Swin, SBIK),

MAD_{bits}=[Log_2(Min_{MAD}x255)]

If $MAD_{Thresold}$>=MAD_{bits} Then

Extract motion vector,

Mv=postion of min_{MAD},

Go to step 5,

Endif

STEP2:

Initializing the MAD counter:Let Count=1.

Compute the MAD of the Y-axis odd blocks,

For i=(Y_{cnter} div 2) to (S_C-$JUMP_i$)*$stepsize_{threshold}$ step $jump_i$

MAD-array(Count)=MAD(Y_{cnter},i, SWIN,SBLK),

Increment count by 1

EndFor

STEP3:

Compute the MAD of the X-axis even blocks.

For j=(X_{cnter} div 2) to (S_r-$Jump_j$)*$StepSize_{threshold}$ $StepJump_j$

MAD_array(Count)=MAD(j,X_{cnter},Swin,SBIK).

Increment Count by I,

Endfor

STEP4:

Find the minimum MAD value form MAD array(MAD_array)

$Min_{MAD=}$Min(MAD-$_{arrray}$).

Comparing the Min_{MAD} with MAD $_{threshold}$ and find the motion vector.

$MAD_{bits=}$[Log_2(Min_{MAD}x255)]

If $MAD_{Thresold}$ >=MAD_{bits} Then

Extract motion vector,

MV=position of Min_{MAD} ,

Go to STEP5 ,

Else

Changing the location of X_{cnter} and Y_{cnter} according to Min_{MAN} location.

Y_{cnter}=The location Y of Min_{MAD},

X_{cnter}=The location X of Min_{MAD},

Detecting the new values of the X_{enter} and Y_{center} if they are equal or exceed The search window boundaries,

If ((X_{cnter}+$Jump_i$)>=Sc)OR((X_{cnte}r<=1)) Then

X_{cnter}=the Left or Right boundary value,

$Jump_i$=$_{jumpi}$ div2,

EndIF.

If(Y_{cnter}+$jump_j$>=sr) or (Ycntr<=1) then

$Jump_j$=$jump_j$ div 2;

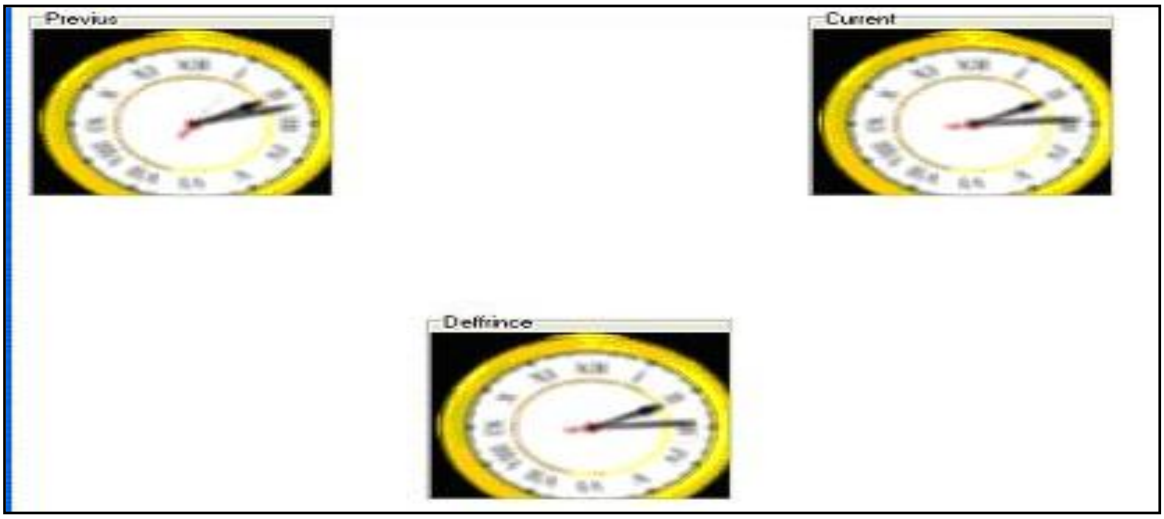
Endif.

Repeat the steps2,3 and 4 until finding appropriate motion vector,

EndIf.

STEP5:

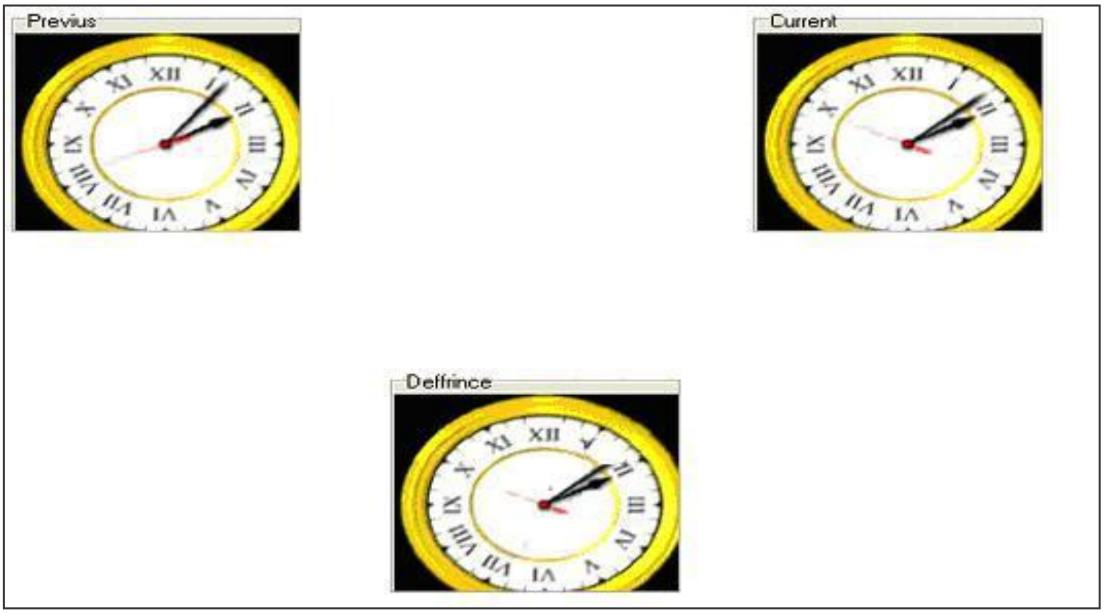
END.



figure(4):Some reconstructed frames for the Clock video sequence using the proposed(oddeven search) and threshold-qurater-pixel.

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D Logarithmic search	62.12	34.75	6330	0:1:49
One time search	62.07	34.50	6330	0:0:48
Proposed(odd-even search	64.44	34.58	6324	0:0:15

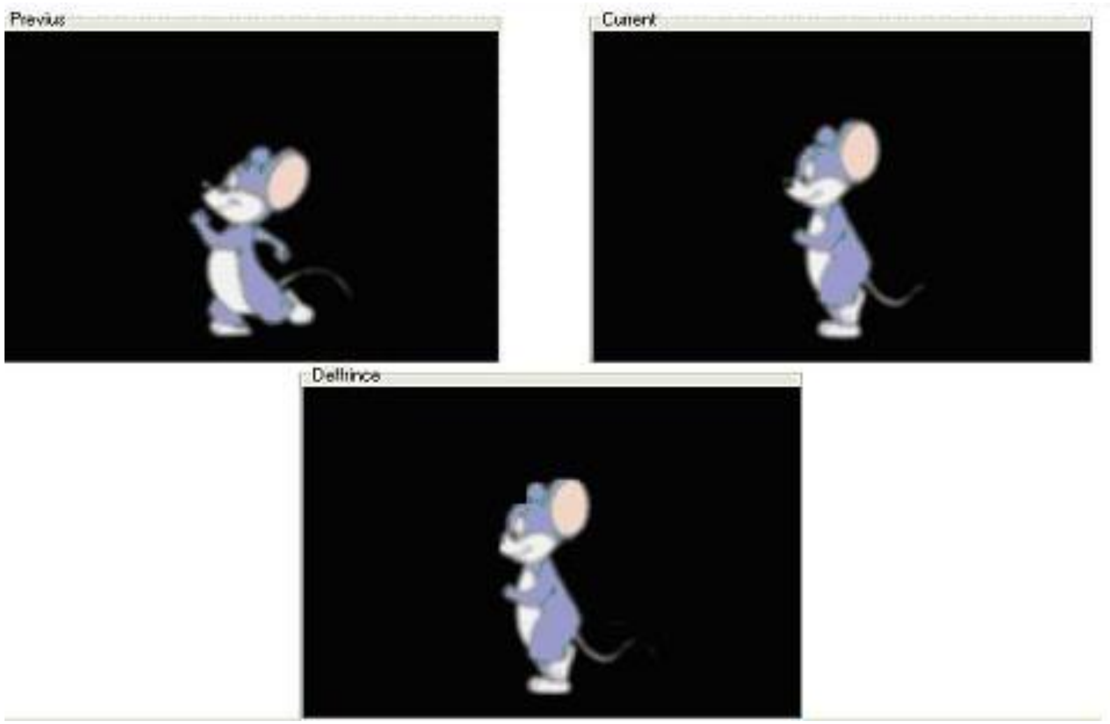
Table(1)clock image with threshold=4



Figure(5): Some reconstructed frames for the Clock video sequence using the proposed(oddeven search) and half-pixel.

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D Logarithmic search	52.74	35.19	6422	0:1:35
One time search	51.57	35.53	6506	0:0:40
Proposed(odd-even search	52.76	35.10	6446	0:0:15

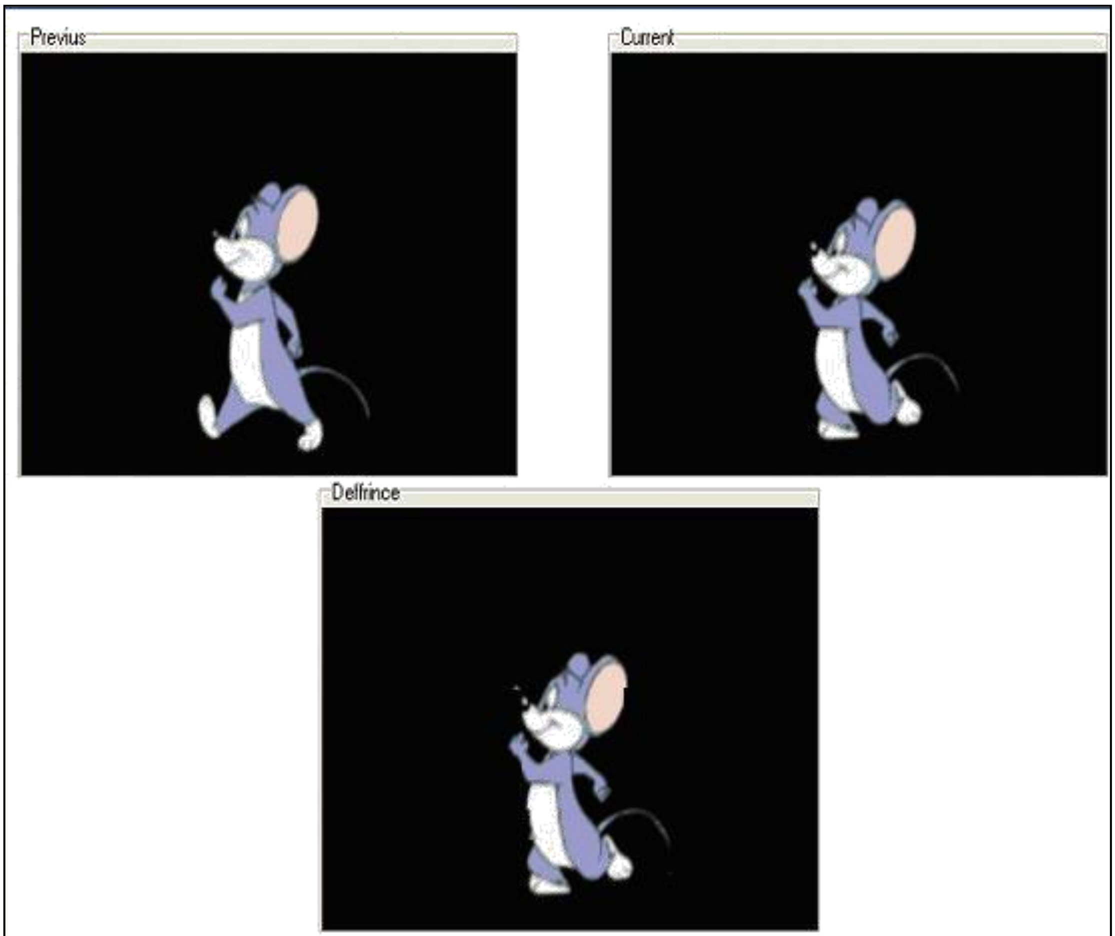
Table(2)clock image with half-pixel



Figure(6): Some reconstructed frames for the mouse video sequence using the proposed(oddeven search) and threshold-qurater-pixel.

Type of motion estimation	**Overall**			
	Cr	**PSNR**	**NB**	**Search time**
2-D Logarithmic search	**81.03**	**45.75**	**6834**	**0:1:50**
One time search	**80.14**	**46.08**	**6864**	**0:0:56**
Proposed(odd-even search	**84.74**	**45.59**	**6834**	**0:0:19**

Table(3) image with threshold=4



Figure(6): Some reconstructed frames for the mouse video sequence using the proposed(oddeven search) and half-pixel.

Type of motion estimation	Overall			
	Cr	PSNR	NB	Search time
2-D Logarithmic search	71.03	42.75	7234	0:1:48
One time search	70.14	46.08	7239	0:0:46
Proposed(odd-even search	71.74	42.59	7230	0:0:15

Table(4)clock image with half-pixel

5-Discussion

The idea discussed in this paper implies developing several improvement on the parameters that H.263 CODEC efficiency. Particularly image accuracy and encoding delay time.

These improvements included thresholding quarter-pixel technique which develops the traditional Half-pixel technique to enhance video frame accuracy through computing values indicate color distance relationship of a group of pixels. it can be noticed that:

1-Using both the proposed OES and thresholding quarter-pixel resulted in noticeable in PSNR of the traditional technique which is characterized with a high PSNR value.

2-With increasing PSNR using the two proposed techniques (Thresholding quarter-pixel, OES). A high compression ratio(Cr) was maintained in comparison with 2-D logarithmic and 3SS techniques

6-Conclusion

From the results obtained of applying two algorithms have been developed: thresholding quarter-pixel technique and OddEven technique.these algorithms have enhanced video compression. The idea of the first algorithm is to increase the image clarity and accuracy, while the second one is concerned with decreasing the elapsed time of video compression and increasing the compression ratio.

Testing these two techniques gave good results, improving significantly the H.263 technique performance. We can conclude that:

1 applying the propsed image enhancement technique(thresholding quarter-pixel) resulted in high accuracy and clarity in decompressed image compared with the tradition half-pixel technique

2-The proposed OES technique has remarkably decreased elapsed time of video compression in comparison with the two well-know techniques(conjugate2D-Lograithm and 3SS). Using three different threshold value by OES resulted in fast block matching process, this decreased encoding delay time clearly.

7-Future work

1-developing a new motion compensated temporal filter depending on distance pixel displacement between successive video frames.

2-developing block matching technique to perform specific comparison operations to choose the best search position.

References

1 Andreas,B. and Roni, K.2000. A 3D-DCT Real-Time Video Compression System for Low Complexity SINGLE-Chip VLSI Implementation.Proc. of the MoMuC2000,P.1B-5-1.Tokyo.
2 Hatem, E.,Ismail, F., and Mohamed, H.2002. Intelligent Motion Detection And Tracking System. Pro.Of the 2002 International Arab Conference on Information Technology(ACIT' 2002),University of Qatar, Qatar.
3 Jeffay, K., and Zhang, H.2002. readings in multimedAiaComputing and Networking. Academic Press,USA.
4 Gonzales, R.c., and woods, R.E., "Digital Image processing", Addison Wesley,1992.

الخلاصة

يمكن تعريف الصور الرقمية على انها التمثيل المرئي لبيانات رقمية ذات معنى ضمن حدود البعد المكاني، ويطلق على ترتيب عدد محدد من الصور الرقمية بصور متسلسلة صور متحركة (Image Animated)
او مقطع فيديوي(video) . يعرف الفيديو الرقمي على انه كل المعلومات الصورية التي تخزن وتناقل في شكل رقمي، على سبيل الذكر، مؤسسات البث التلفزيوني الرقمية. ان اكثر الصور الرقمية تحتوي على نسبة عالية من وفرة البيانات وتكرارها حيث يشكل تكرار تلك البيانات عائقا كبيرا اذا ما تم تناقلها عبر وسائل تناقل البيانات المختلفة امثال الشبكات الموسعة والانترنت. لذا اقتضت الحاجة اختزال ما تكرر من تلك البيانات بحيث لا يؤثر ذلك على هيئتها العامة في حالة استرجاع ما اختزل منها باستخدام مجموعة من تقنيات الضغط الصوري الفديوي. تم العمل في هذا البحث على تحسين فعالية تقنية H.263 للضغط الفديوي والتي تختص بتطبيقات التخاطب المباشر بعيدالمدى عبر الانترنت. تمت دراسة خصائص هذه التقنية وتم التعرف على هيكلها العام، اضافة الى ذلك تم تحديد العوامل التي تؤثر على كفاءة هذه التقنية مثل عاملي (الوقت المستغرق في عملية الضغط، وكفاءة ووضوح الصورة الرقمية) من خلال بناء وتحليل مجموعة من خوارزميات توقع الحركة الصورية مثل (conjugate search,(ots),2_D Logarithmic search) التي تستخدم في عملية الضغط الفديوي. اقترح هذا البحث تطوير تقنيتين جديدتين تحسن من كفاءة تقنية H.263 وهما (thresholding quarter_pixel, oddeven search Algorithm) ، حيث خصصت الاولى للعمل على تحسين الخصائص الصورية للاطر الفديوية، اما التقنية الثانية فانها تقلل من الوقت المستغرق في عملية الضغط الفديوي.اوضحت نتائج هذا البحث بشكل مفصل كفاءة التقنيتين المقترحتين مقارنة بالطرق المعروفة والموضحة سابقا من حيث عامل الوقت، الكفاءة الصورية ونسبة الضغط.استنتجت هذه الدراسة ان تقليل عامل الوقت وزيادة نسبة الضغط مع المحافظة على الخصائص الصورية ووضوح الصورة الرقمية للفيديو الرقمي يزيد من كفاءة تقنية H.263.