ADAPTIVE EARLY EXIT TECHNIQUES FOR MINIMUM DISTORTION CALCULATION IN IMAGE CORRELATION

BACKGROUND

Image compression techniques can reduce the amount of data to be transmitted in video applications. This is often done by determining parts of the image that have stayed the same. The "motion estimation" technique is used in various video coding methods.

Motion estimation is an attempt to find the best match between a source block belonging to some frame N and a search area. The search area can be in the same frame N, or can be in a search area in a temporally displaced frame N-k.

These techniques may be computationally intensive.

BRIEF DESCRIPTION OF THE DRAWINGS

These and other aspects will now be described in detail with reference to the accompanying drawings, wherein:

Figure 1 shows a source block and search block being compared against one another;

Figure 2 shows a basic accumulation unit for measuring distortion;

Figure 3a and 3b shows different partitioning of the calculations among multiple SAD units;

Figure 4 shows a tradeoff between early exit strategy calculations and an actual total calculation;

Figure 5 shows a flowchart of the early exit strategy;

Figure 6a shows an early exit using an early exit flag;

Figure 6b shows early exit using a hardware status register;

Figure 7 shows a flowchart of operation of the adaptive early exit strategy.

DETAILED DESCRIPTION

Motion estimation is often carried out by calculating a sum of absolute differences or "SAD". Motion estimation can be used in many different applications, including, but not limited to cellular telephones that use video, video cameras, video accelerators, and other such devices. These devices can produce video signals as outputs. The SAD is a calculation often used to identify the lowest distortion between a source block and a number of blocks in a search

region search block. Hence the best match between these blocks. One way of expressing this is

$$SAD = \sum_{i=0}^{N-1} \sum_{j=0}^{n-1} |a(i,j)-b(i,j)|, \quad N = 2,4,8,16,32,64.$$

Conceptually what this means is that a first frame or source block (N) is divided into component parts of MxN source blocks 100. These are compared to a second frame (N-K) 102. The frames can be temporally displaced, in which case $k \neq 0$. Each N-K frame 102 is an $M+2m_1$ x $N+2n_1$ area. The source block 100 is shown in the center of the area in Fig. 1. The parts of the images that match can be detected by correlating each part of each image frame against other image frame using the distortion measurer. The compression scheme uses this detection to compress the data, and hence send less information about the image.

This device can also be part of a general-purpose DSP. Such a device is contemplated for use in video camcorders, teleconferencing, PC video cards, and HDTV. In addition, the general-purpose DSP is also contemplated for use in connection with other technologies utilizing digital signal processing such as voice processing used in mobile telephony, speech recognition, and other applications.

The speed of the overall distortion detection process can be increased. One way is by using hardware that allows each SAD device to carry out more operations in a cycle. This, however, can require more expensive hardware.

Another way is to increase the effective pixel throughput by adding additional SAD devices. This can also increase cost, however, since it requires more SAD devices.

Faster search algorithms attempt to use the existing hardware more effectively.

The block SAD compares the source group against the "search group". The source group and the search group move throughout the entire image so that the SAD operation calculates the overlap between the two groups. Each block in the source group will be compared to multiple blocks in each of the search regions.

A typical SAD unit operates on two, 16 by 16 elements to overlay those elements on one another. This overlay process calculates 16 X 16 = 256 differences. These are then accumulated to represent the total distortion.

The SAD requires certain fundamental operations. A difference between the source X_{ij} and the search Y_{ij} must

be formed. An absolute value $|X_{ij}-Y_{ij}|$ is formed. Finally, the values are accumulated, $SAD=\sum_{i=0}^{N-1}\sum_{j=0}^{n-1}|X_{ij}-Y_{ij}|$.

A basic accumulation structure is shown in Fig. 2 Arithmetic logic unit 200 receives X_{ij} and Y_{ij} from data buses 198,199 connected thereto, and calculates $X_{ij}-Y_{ij}$. The output 201 is inverted by inverter 202. Both the inverted output, and the original, are sent to multiplexer 204 which selects one of the values based on a sign bit 205. A second arithmetic logic unit 206 combines these to form the absolute value. The final values are stored in accumulation register 208. Effectively, this forms a system of subtract, absolute, accumulate, as shown in Figure 2.

Figure 2 shows a single SAD computation unit. As noted above, multiple computation units could be used to increases the throughput. If the number of computation units is increased, that increases, in theory, the pixel throughput per cycle.

The present inventor noted, however, that increase in pixel throughput is not necessarily linearly related to the number of units. In fact, each frame is somewhat correlated with its neighboring frames. In addition, different parts of any image are often correlated with other parts of the image. The efficiency of the

compression may be based on characteristics of the images. The present application allows using the multiple SAD devices in different modes, depending on the efficiency of compression.

The present application uses the architecture shown in Figures 3A and 3B. The same connection is used in both Figures 3A and 3B, but the calculations are partitioned in different ways.

Figure 3A shows each SAD device 300, 302 being configured as a whole SAD. Each SAD receives a different block, providing N block SAD calculations. Effectively, unit 301, therefore, calculates the relationship between a 16 by 16 reference and a 16 by 16 source, pixel by pixel. Unit 2, 302 calculates the result the difference 16 by 16 source and the 16 by 16 search pixel by pixel. The alternative shown in Figure 3B. In this alternative, configuration each single SAD 300, 302 performs a fraction of a single block SAD calculation. Each of the N computation units provides 1/N of the output. This "partial SAD" operation means that each of the 8 bit subtract absolute accumulate units have calculated 1/N of the full SAD calculation configured to that unit.

The overall system that determines the whole or partial should be used based on previous results as

described herein. This in turn can reduce the number of calculations that is carried out.

One way to determine whether whole or partial is used is to assume that temporally close images have correlated properties. A first cycle can be calculated using the whole SAD mode, and a second cycle can be calculated using the partial SAD mode. The cycle which works faster is taken as the winner, and sets the SAD mode. This calculation can be repeated every X cycles, where X is the number of cycles after which local temporal correlation can no longer be assumed. This can be done in a logic unit, which carries out the flowchart of Figure 7, described herein.

Throughput can also be increased by an "early exit" technique as described herein.

The complete SAD calculation for 16x16 elements can be written as $|p_1r - p_1s| + |p_2r - p_2s| + \ldots |p_{256}s - p_{256}r|$...(1). If all of these calculations were actually carried out, the calculation could take 256/N cycles, where N is the number of SAD units. It is desirable to stop the calculation as soon as possible. Interim results of the calculation are tested. These interim results are used to determine if enough information has been

determined to find a minimum distortion. The act of testing, however, can consume cycles.

The present application describes a balance between this consumption of cycles and the determination of the minimum distortion. Figure 4 illustrates the tradeoff for a 16x16 calculation using 4 SAD devices. Line 400 in Figure 4 represents the cycle count when there is no early exit. The line is horizontal representing that the cycle count without early exit is always 256/4=64. The cycle counts for early exit strategies are shown in the sloped lines 402, 404, 406 and 408. Line 404 represents one test every sixteen pixels, line 406 represents one test every thirty-two pixels (1/8) and line 408 represents one test every sixty-four pixels (1/16). Note that when the lines 402-408 are above line 400, the attempt at early exit has actually increased the overall distortion calculation time. Line 402 represents the cycle consumption where zero overhead is obtained for exit testing. That is, when a test is made, the exit is always successful. Line 402 is the desired goal. An adaptive early exit scheme is disclosed for doing so.

Block I is first processed using any normal strategy known in the art to find a minimum distortion. This can be done using test patterns, which can be part of the

actual image, to find the distortion. This minimum distortion is used as the baseline; and it is assumed that block I + n, where n is small, has that same minimum distortion. Two basic parameters are used.

Kexit(N) represents the number of pixels that have been processed previously for a search region before an early exit is achieved.

Aexit(N) represents the state of the partial accumulator sign bits, at the time of the last early exit for a search region.

For these blocks I + n, the SAD calculation is terminated when the distortion exceeds that threshold. This forms a causal system using previous information that is known about the search region.

The usual system is based on the image characteristics within a search region being some probability of maintaining common characteristics from time to time. The time between frames is between 1/15 and 1/30 of second, often fast enough that minimal changes occur during those times above some noise floor related to measurable system characteristics. Also, there are often regions of an image which maintains similar temporal characteristics over time.

According to the present application, the accumulator unit for each SAD can be loaded with the value (- least/n), where "least" represents the minimum distortion that is measured in the block motion search for the region. Many SAD's are calculated for each search region. The first SAD calculating for the region is assigned the "Least" designation. Future SADs are compared to this, to see if a new "Least" value has been established. When the accumulators change sign, the minimum distortion has been reached. Moreover, this is indicated using only the existing SAD structure, without an additional calculation, and hence additional cycle(s) for the test.

A test of the character of the image can be used to determine how many of the accumulators need to switch before establishing the early exit. For example, if source and target regions are totally homogeneous, then all the accumulators should change sign more or less at the same time. When this happens, any one of the running SAD calculations exceeding the previous least measurement can be used to indicate that an early exit is in order.

This, however, assumes total image homogeneity. Such an assumption does not always hold. In many situations, the multiple accumulators of the different

SAD units will not be increasing at the same rate. Moreover, the different rate of increase between the accumulators may be related directly to the spatial frequency characteristics of the differences themselves, between the source and target block, and also to the method of sampling the data. This can require more complex ways of considering how to determine early exit, based on what happens with the SAD units.

One operation is based on the probability associated with a split SAD state; where not all of the SAD units are in the same state. This difference in rate of increase between the accumulators is related to the spatial frequency characteristics of the difference between the source and target block. Since these spatial frequency characteristics are also correlated among temporally similar frames, the information from one frame may also be applied to analysis of following frames.

This is explained herein with reference to variables - where A_1, A_2, A_3 … A_n are defined as events associated with a split SAD calculation.

The events can be defined as follows:

Event $A_i = SAD_i \geq 0$ where $SAD < 0$ for $i \neq j$.

This conceptually means that the event A_i is defined as occuring when SAD unit i is positive and all the

remaining SAD units are negative. This would occur, for example, when the accumulators were increasing at different rates. This can also be defined as combined events, specifically:

Event $B_{i,j} = A_i \cup A_j = SAD_i \geq 0$ for $SAD_j \geq 0$, and where $SAD_k < 0$ for $k \neq i, j$. This means that event $B_{i,j}$ is defined as "true" when A_i exists and A_j are true, but all other A_k are false. The concept of defining the operations in terms of events can be extended to include all the possible combinations of i, j and k. This yields, for 4 SAD units, a total of 16 combinations. For larger numbers of SAD units, it leads to other numbers of combinations, and possibly using more variables, such as i, j, k and m or others.

Describing this scenario in words, each event "B" is defined as the sum of the specified accumulators being greater than 0. Each of these combinations is defined as a probability. For 4 SAD units, there are total of 16 possible states of accumulators. These can be grouped according to how they are handled.

A first trivial possibility is

$$P\left(B \middle| \bar{A}_1 \cap \bar{A}_2 \cap \bar{A}_3 \cap \bar{A}_4\right) = 0.$$

This means that the probability that sum of the accumulators is > 0, given that none of the accumulators has exceeded 0, is 0.

The opposite is also true:

$$P\ (B|A_1 \cap A_2 \cap A_3 \cap A_4) = 1;$$

Which means that the probability of the sum of all the accumulators is set, given that none of them are set, is also 1.

Excluding these trivial characteristics, there are 14 nontrivial combinations. The first group includes four cases where one of the accumulators is set and the remaining three are not set:

$$P(B|\ A_1 \cup (\bar{A}_2 \cap \bar{A}_3 \cap \bar{A}_4),$$

$$P(B|\ A_2 \cup (\bar{A}_1 \cap \bar{A}_3 \cap \bar{A}_4),$$

$$P(B|A_3 \cup (\bar{A}_1 \cap \bar{A}_2 \cap \bar{A}_4),$$

$$P(B|A_4 \cup (\bar{A}_1 \cap \bar{A}_2 \cap \bar{A}_3).$$

Another group represents those conditions where two of the accumulators are set, and the other two accumulators are not set. These combinations are written as:

$P(B|A_1 \cap A_2) \cup (\bar{A}_3 \cap \bar{A}_4)$

$P(B|A_1 \cap A_3) \cup (\bar{A}_2 \cap \bar{A}_4)$

$P(B|(A_1 \cap A_4) \cup (\bar{A}_2 \cap \bar{A}_3)$

$P(B|A_2 \cap A_3) \cup (\bar{A}_1 \cap \bar{A}_4)$

$P(B|A_2 \cap A_4) \cup (\bar{A}_1 \cap \bar{A}_3)$

$P(B|A_3 \cap A_4) \cup(\bar{A}_1 \cap \bar{A}_2)$

Finally, the following group represents the cases where three accumulators are set and one accumulator is not set

$P(B|A_1 \cap A_2 \cap A_3) \cup \bar{A}_4)$

$P(B|A_2 \cap A_3 \cap A_4) \cup \bar{A}_1)$

$P(B|A_1 \cap A_3 \cap A_4) \cup \bar{A}_2)$

$P(B|A_1 \cap A_2 \cap A_4) \cup \bar{A}_3)$.

The present embodiment recognizes that each of these groups, and in fact each of these situations, represents a different condition in the image. Each group or each situation can be handled differently.

This system operates as above, and as described with reference to the flowchart of Figure 5. The final goal is to complete the calculation, and hence to exit, sooner. This is shown in Fig 5 by first, determining matching characteristics of two images; a source image and a search image at 550. The matching characteristics

are calculated without any early exit. The minimum distortion is found at 555 and the conditions when that minimum distortion existed are found at 560.

The conditions at 560 can include a grouping type that existed at the time of minimum distortion, or the specific condition among the 14 possibilities.

At 570 a subsequent image part is tested. This subsequent part can be any part that is correlated to the test part. Since temporally correlated images are assumed to be correlated, this can extend to any temporally correlated part.

The image source and search are tested, and a determination of the specific groupings that occurred at the time of minimum distortion is found at 575. An early exit is then established, at 580.

The early exit, once determined, can be carried out in a number of different ways.

Figure 6a shows a system of carrying out the early exit using an early exit or "EE" flag. N SAD units are shown, where in this embodiment, N can be 4. Each SAD unit includes the structure discussed above, and specifically ALUs, inverters, and accumulators.

The output of each of the accumulators is coupled to a combinatorial logic unit 600 which arranges the

outputs. This can be used to carry out the group determination noted above. The combinatorial logic unit is carried out using discrete logic gates, e.g., defined in hardware definition language. The gates are programmed with an option based on the selected group. Different images and parts may be processed according to different options.

For each option, the combination of states, e.g., the group discussed above, is coded. The combinatorial logic monitors the accumulators of all the SAD units. Each state is output to a multiplexer.

When those accumulators achieve a state that falls within the selected coding, an early exit flag is produced. The early exit flag means that the hardware has determined an appropriate "fit". This causes the operation to exit.

Figure 6B shows an alternative system, in which the states of the accumulators are sensed by a hardware status register 600. The status register is set to a specified state by the condition of the accumulators. The status register stores the specified condition that represents the early exit. When that specified condition is reached, the early exit is established.

The way in which the adaptive early exit is used, overall, is described in reference to Figure 7. At 700, the video frame starts. 705 represents buffering both frame M and frame M+1. 710 is a determination if the block history model needs update. This can be determined by, for example, monitoring of the time since a previous frame update. For example, x seconds can be established as a time before a new update is necessary.

If the model needs updating, then the process continues by loading the accumulators with 0xFF01 and setting the local variable N=1 at 715. At 720, the system obtains SAD search region N and uses the periodic exit test T_{exit} =1/16..., at step 725 the exit test is performed. If successful, a local variable Kexit(N), which is the pixels before exit and Aexit(N) which is an summary of accumulators 1 through 4 before exit restored. The local variable n is also incremented at step 730. This establishes the local parameters, and the process continues.

In a subsequent cycle the block history of update does not need to be redone at step 710, and hence control passes to step 735. At this step, the previously stored Kexit and AEexit are read. This is used as the new count at step 740 to set target block flags.

At step 745, a search for block N is established, an a exit and Kexit are updated at step 750. N is incremented. At step 755, a determination is made whether N is equal to 397. 397 is taken as the number of frames in the buffer, since there are 396, 16x16 blocks in a 352x288 image. However, this would be adjusted for different size sizes as applicable.

Again, the temporal variations of large portions of an image are likely to remain unchanged. Therefore, when the partial accumulators have a specific sign bit, their state produces significant advantages. Moreover, the time between frames is usually on the order of 1/15 to 1/30 of a second. Finally, regions within the image maintain their localized characteristics, and therefore their spatial frequency may be correlated.

Although only a few embodiments have been disclosed, other modifications are possible.

What is claimed is:

1. A method comprising:

determining a minimum distortion between two regions;

loading a value related to said minimum distortion into an accumulating element;

accumulating results of subsequent image distortion calculations into said accumulating element; and

establishing said minimum distortion for said subsequent calculation exit when said accumulating element reaches a predetermined value.

2. A method as in claim 1, wherein there are a plurality of said accumulating elements, and said loading comprises loading said value into each of said accumulating elements.

3. A method as in claim 2 wherein said establishing occurs when specified accumulating elements of said plurality of accumulating elements reach said predetermined value.

4. A method as in claim 2 wherein said establishing occurs when all of said plurality of accumulating elements reach said predetermined value.

5. A method as in claim 2 wherein said establishing occurs when a single accumulating element reaches said predetermined value.

6. A method as in claim 2 wherein said value related to said minimum distortion is an amount of distortion, divided by a number of said accumulating elements.

7. A method as in claim 2 further comprising determining a character of the regions, and determining, from said character, a number of accumulating elements to require to reach said predetermined value before said establishing.

8. A method as in claim 7, wherein said character of the image indicates a homogeneous image, and said establishing said minimum distortion is based on results from a single accumulator.

9. A method as in claim 7, wherein said establishing comprises establishing said early exit based on results from plural accumulators.

10. A method as in claim 9 wherein said early exit comprises determining probabilities of each of the accumulators establishing certain states, and said establishing is based on said accumulators establishing said certain states.

11. A method as in claim 6, wherein said predetermined value is zero.

12. A method comprising:

determining a distortion between two image regions; and

carrying out an early exit scheme without using an extra calculation cycle for the early exit scheme.

13. A method as in claim 12 wherein said early exit is calculated by loading a value into an accumulator element and changing said value based on said determining.

14. A method as in claim 13, wherein there are a plurality of said accumulator elements, and said loading comprises loading said value into each of said accumulator elements.

15. A method as in claim 14 wherein said establishing occurs when specified accumulator elements of said plurality of accumulator elements reach a predetermined value.

16. A method as in claim 14 wherein said establishing occurs when all of said plurality of accumulator elements reach said predetermined value.

17. A method as in claim 12 wherein said value is related to a minimum distortion in a previous calculation.

18. A method as in claim 13 further comprising determining a character of the regions, and determining, from said character, a number of accumulator elements to require to reach said predetermined value before said establishing.

19. A method as in claim 18, wherein said character of the image indicates a homogeneous image, and said establishing said minimum distortion is based on results from a single accumulator element.

20. A method as in claim 15, wherein said value is zero.

21. A method of determining distortion between image blocks comprising:

determining a first distortion between a first image part and a second image part, and obtaining a count of calculations for an early exit and state of certain calculation at the time of the early exit; and

in a subsequent calculation, terminating a calculation based on both said count and said state.

22. A method as in claim 21 wherein said distortion is calculated by loading values into accumulators and establishing an early exit when said accumulators reach a specified value.

23. A method as in claim 22, wherein said specified value is zero.

24. A method of processing image parts comprising;

obtaining image distortion information about a first image part; and

using said image distortion information to process a second part, by loading a value related to said image distortion information into a calculating element, and terminating a calculation performed by said calculating element when said value reaches a predetermined amount.

25. A method as in claim 24 wherein said image distortion information includes a minimum of distortion measured in a block motion search.

26. A method as in claim 24 wherein said image distortion is used to calculate a distortion in a sum of absolute differences device with an accumulator.

27. A method as in claim 26 wherein said value is loaded into the accumulator.

28. A method as in claim 18 wherein said value comprises (-least/n) where least is a value found in a previous cycle and n is a number of accumulators.

29. A method as in claim 28 wherein said predetermined amount is 0.

30. A method as in claim 18 wherein said accumulators are loaded with a said value (-least/ n) where least is an expected early value and n is a number of accumulators, and wherein an early exit is established when an accumulator reaches 0.

31. A method as in claim 22 wherein said value represents sign bits of the accumulators.

32. A method, comprising;

obtaining a plurality of image manipulating devices which achieve specified states based on relationships between two images;

accumulating results based on said relationships, and

determining whether a calculation is completed based on the states of the accumulating.

33. A method as in claim 32, wherein said image manipulating device is part of a video camera.

34. An apparatus, comprising:

a calculating device, including an accumulator element, that calculates values related to a minimum distortion between two regions;

a buffer, storing a value related to said minimum distortion, and loading said distortion into said accumulator; and

a logic unit, establishing a minimum distortion based on a value in said accumulating device.

35. An apparatus as in claim 34, wherein there are a plurality of said calculating devices, and said distortion is loaded into each of said accumulating elements.

36. An apparatus, comprising:

a video acquiring device;

a plurality of calculating devices, each including an accumulator element, that calculates values related to a minimum distortion between two video portions acquired by said video acquiring device;

a buffer, storing a value related to said minimum distortion, and loading said distortion into said accumulators; and

a logic unit, establishing a minimum distortion based on values in said accumulating device.

37. An apparatus as in claim 36 wherein said logic unit monitors sign bits of said accumulator elements.

38. An apparatus as in claim 36 wherein said logic unit monitors a value in said accumulator elements.

39. An apparatus as in claim 36, wherein said video acquiring device is a video camera.

40. An apparatus as in claim 38, wherein said logic unit monitors a value in all of said plurality of

accumulating elements to reach said predetermined value.

41. An apparatus as in claim 36, wherein said accumulators are loaded with a value (-least/n) where least is an expected early value and n is a number of accumulators, and wherein an early exit is established when an accumulator reaches 0.