A Stochastic Diffused Motion Estimation Method for h.264/AVC Encoder in Telemedicine using VLSI

Nandhakumar A.* and Senthilkumar A.
Department of EEE, Dr. Mahalingam College of Engineering and Technology, Pollachi, INDIA
*nandhu.udt@gmail.com

Abstract
In telemedicine videoconferencing technologies using internet becomes crucial due to the extensive reach of wireless technologies. In this scenario encoding an decoding plays a very crucial role to save bandwidth. Motion estimation is a significant component in digital video compression / transmission. Conversely, it is found to be difficult to implement real-time codec due to its natural computational complexity. The past years have witnessed investigations on rapid block-matching protocols to considerably decrease the computation complexity in motion estimations. These algorithms had created a massive awareness and led to many more promising algorithms for VLSI systems to reduce computational intricacy. Despite, block matching algorithms having easy implementation process, it leads to suboptimal solutions if there is no full search being carried out. The computational cost for full search Block Matching Algorithm is so high, thereby preventing it from being used in VLSI systems. This work suggests innovative fast motion estimation derived from stochastic diffused search protocol for motion estimation.

Keywords: Motion estimation, Stochastic diffused search algorithm, Block Matching Algorithm, VLSI systems, video compression.

Introduction

Normally, video series have a considerable quantity of statistical as well as subjective redundancy in and among frames. The biggest goal of video source coding is to reduce bit-rate in order to store and transmit by investigating both statistical as well as subjective redundancies. It is also aimed at encoding a “minimum set” of data by employing entropy coding methods. Consequently, the coded video data gets compressed in contrast to the original source information. The functional capabilities of video compression methods are mostly dependent on the quantity of redundance enclosed in the image data and also on the actual compression methods utilized for coding [1]. The practical coding methods are used in exchange between coding performance as well as implementation complexity. Actually, all video coding methods that were standardized in the past were built as well as optimized by considering the potentials of the modern VLSI technology.

The fundamental aim of video coding is bit-rate reduction which is achieved by eliminating redundant data. The coding schemes are classified into source as well as entropy coding. The source materials are used in source coding where the yielded results are lossy, meaning degraded picture quality. The source coding is again divided into intra as well as inter-frame coding. Intra-frame coding is employed in primary picture as well as for subsequent pictures that occur with change of scene. Inter-frame coding is used for series of analogous pictures that includes dynamic objects [2]. Among the vital aspects of video coding, the most significant one is to define a data structure enabling the decoder to decode the received bit-stream without any uncertainty.

Compression is a procedure in which the size of the dataset is reduced which in turn reduces the bandwidth that is required to represent the signals digitally. Using compression technique it is possible to minimize transmission time as fewer data is being forwarded. Further it also reduces storage requirements. Through compression it is possible to reduce the size of text files, pictures, voice, video and also redundant data [3]. At present, there are numerous video coding models like MPEG-2, MPEG-4, H.264/AVC, VC-1, AVS as well as HEVC. The video codec that supports several standards are more efficient when compared to other inter-communications among the video devices that uses different other standards.

H.264/ Advanced Video Coding (AVC) are industry models used to compress videos. The H.264/AVC provides improved compression competence as well as and better flexibility in compressing, transmitting as well as in storing video. H.264/AVC is capable of delivering improved image quality together with compressed bit-rate or in other words it delivers similar image quality at a lower bit-rate when compared to other models like MPEG-2 as well as MPEG-4 visual. H.264 is largely employed in video application, for instance in HDTV, even in mobile videos as well as internet applications. This is a standard that was built by the ITU-T VCEG along with ISO [4]. Hardware H.264 encoder may be an ASIC or FPGA that is a normal programmable chip.

A video can be compressed using an encoder while it can be converted back to its original format using a decoder. H.264 video encoder performs predictions, transforms as well as encoding process to generate compressed H.264 bit streams. A H.264 Video Decoder performs the reverse procedure of decoding, inverse transforms as well as reconstructions to generate video series. H.264/AVC model permits 7 modes with adjustable block sizes.16x16pixel macro block may be split into tiny blocks to result in improved compression efficacy.

The rising video resolutions as well as the demand for real-time encoding necessitate fast processors.

The encoder is structured as a pipeline architecture that process data based on MB. There are four phases in pipeline. As depicted in figure 4, the first stage is mode of the sub-pixel motion estimation (ME), the subsequent stage includes the integer ME as well as fraction ME, the third stage has motion compensation (MC) , intra prediction , the TQ as well as Inverse TQ, the final stage includes the De-blocking Filter as well as Entropy code. Finally, the code stream data are the results [5.]

The H.264/ AVC codec is largely employed in video application, for instance in HDTV and even in mobile videos as well as internet application. The rising video resolution as well as the need for real-time encoding necessitate fast processors. The H.264/AVC encoder is made of 3 stages namely predictions, transformations/quantizations as well as entropy encoding. Motion estimations refers to a section of the predictions phase. Figure 1 demonstrates the H.264/AVC encoder [6.]

Motion estimation methods are the basis for H.264/AVC videos compression as well as videos processing application. It retrieves motion data by using the video series in which the motion is generally denoted utilizing a motion vector (x, y). This data is employed in videos compression to identify a suitable matching block with reference to frame in order to measure less energy residue to produce temporal interpolated frames [7]. Nowadays, several motion estimation methods are becoming handy and among them are pel-recursive techniques that obtain motion vectors for all pixels while phase plane correlation method produce motion vectors through correlation among current as well as reference frame. Among the existing techniques the prevalent method is Block Matching methodology.

The concept following block matching is that the existing frame is split into a matrix of ‘macro blocks’, which are later contrasted with associated block as well as its nearby neighbours in the preceding frame to develop a vector, which specifies the movement of a macro block [8].

The search area for an adequate macro block match is restricted by search variable ‘p’ pixels on every side 94) of the respective macro block in previous frame. Bigger motions demand a greater p and if the search variable is larger, then the computational cost becomes higher for motion estimation procedure. Generally, the macro block is considered the square of side16 pixels while the search variable p is 7 pixels.

The block matching algorithm (BMA) consists of two parts namely matching criterion as well as searching strategy. Sum of absolute difference (SAD) as well as full search (FS) are selected for matching condition as well as for search scheme, correspondingly. SAD may be stated as below [9.]

$$SAD(dx,dy) = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left|a(i,j) - b(i+dx, j+dy)\right|$$

$$(MV_x, MV_y) = (\mathrm{dx}, \mathrm{dy})\Big|_{\min SAD(dx,dy)}$$

$a(i,j)$ as well as $b(i,j)$ represents the pixels of reference as well as candidate blocks, correspondingly. *dx* as well as *dy* denotes displacement of the candidate block within the search window. *MxN* refers to the dimensionof the reference block while (MV_x, MV_y) represents the motion vector pair of the block.

This work suggests new fast motion estimation derived from stochastic diffused search algorithm for motion estimation. Section 2 reviews the literature for related works. Section 3 elucidates the suggested methodology. Section 4 shows the simulation result while Section 5 presents the conclusion to the proposed work.

Literature Review

Pastuszak[10] introduced the H.264/AVC encoder infrastructure that can gain from RDO as well as attain high throughputs. The investigation is carried out at the partition as well as macro block levels. The high-throughput simple rates estimation presents minor quality loss as well as backs both entropy coding techniques that exists in H.264/AVC.

The average quality for the 2 coding techniques is reduced by 0.07 as well as 0.17 dB in contrast to JM17 software that uses the RDO. The efficacy of compression may be exchanged for throughput. It may assist up to 1080p at 60 frames per second encoding for 90-nm TSMC technology.

The modern lower cost texture-synthesis based video coding method was offered by Sun et al., [11] for H.264/AVC. It is suitable for a particular set of stochastic textures such as water, cloud and so forth.

The suggested system preserves the texture's higher-order data against the global motions. The sequential texture regions are replicated into the synthesis regions in a significant scale when compared to the current synthesis technique employed in video coding. Thus, the high computation cost is moved to the encoder that may largely make the decoder design simpler.

Li et al[12] suggested a VLSI design for all-intra scalable video encoders to aspire for effective scalable videos encoding. The execution outputs reveal that the suggested SVC encoder may compute greater than 594-k MBs every second. This is equal to the sum of 60 high-definition, 1080-p, SD 480-p as well as frequent intermediary format frames below 135-MHz working frequency. The suggested design utilizes 258-K gate count when synthesized by 90-nm CMOS methodology.

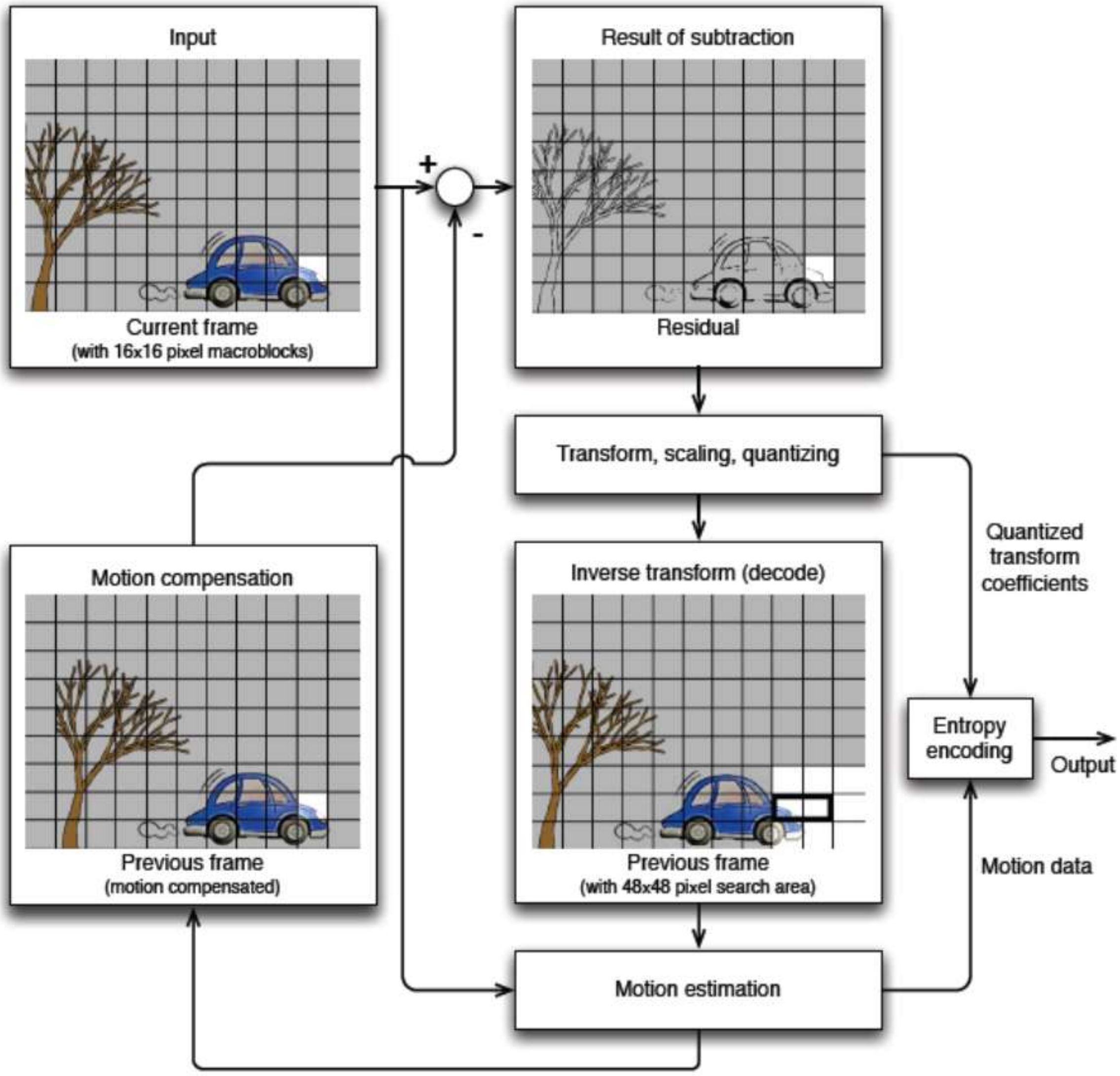

Figure 1: Illustration of the H.264/AVC encoder

Pastuszak and Jakubowski[13] illustrated the adaptive computationally scalable motion-estimation protocol as well as its hardware implementation permits the H.264/AVC encoder to attain effectiveness near optimum in real-time situations. The protocol uses various search schemes to accustom to local motion activities. The quantity of verified search points is established by the encoder controller for every macro-block. The protocol may attain outputs that are near optimal even if the quantity of search points designated to macro blocks is clearly restricted as well as varies with time. The infrastructure employs a new data-flow. The architecture is validated in the real-time field-programmable gate array hardware encoder. The synthesis outputs as well as the real-time validation illustrate that the design is capable of assisting HDTV at 200 MHz for 0.13-μm TSMC technology.

Liu et al[14] used Adaptive Block-size Transform (ABT) idea to FME design as well as suggested a full-mode FME infrastructure on the basis of 8×8/4×4 adaptive Hadamard Transform. This method may shun from combining all flexible block-size blocks into 4×4-size blocks as well as enhance the encoding performance. Such approaches may amplify parallelism as well as minimizes the cycles effectively. The simulation outputs demonstrate that the suggested design is capable of achieving a real-time processing for QFHD@30fps.

Correa et al[15] suggested the Stationarity, Heterogeneity as well as Border Strength (SHBS) heuristic to minimize the quantity of computations required to evaluate the suitable inter-frame mode. By applying SHBS, there is a cutback of 168 times in the encoding iterations, with an improved PSNR, at the expense of a fairly minimal bit-rate increase.

Methodology

The objective is to find an efficient encoder for telemedicine application and simulations were carried out using medical video frames using the Stochastic Diffusion Search (SDS).

Stochastic Diffusion Search (SDS): The computing of BM is seeking for a suitable position in the search window, wherein the minima of MAD is to be determined. To reach an improved MAD, the numerous positions in the search window are to be coordinated. Conversely, the increased processing time is spent on searching. An improved matching protocol must be able to expend lesser computational time on search as well as attain improved position [16]. The objective of applying the SDS protocol to ME is the acceleration of matching search as well as reaching a better ME.

Consecutively, the computational difficulty that exists in BM process can be reduced through various BM protocols, which are suggested by contemplating the three methods: (1) use of fixed pattern: the search operation is carried over a preset sub-set of the entire search window. (2) reduce the number of the search points: the positions that reduce the error-function (SADvalues) iteratively are selected by the protocol (3) the computation overheads for all search points is reduced: the matching costs (SAD operations) are substituted using a fractional or an easy version which feature lesser intricacy [17]. Though, only partial pixels enter the matching computations, the usage of such common subsampling methods may significantly influence the precision in detecting the motion vectors that occurs because of noise and illumination change.

SDS presented a modern probabilistic method for solving best-fit patterns recognition as well as matching issues. SDS has a rigid mathematical structure when compared to many other natured inspired search algorithms. This framework illustrates the activities of the protocol by examining its resources allocation, convergence to global optima, toughness as well as minimum convergence condition along with linear time complexity.

For introducing SDS, which is a social metaphor, it is vital to introduce the Mining Game. The metaphor offers an easy higher-level description of the activities of agents in SDS, in which the mountain range is split into hills while every hill is split into regions.

To increase their collective wealth, miners are required to determine the hill that has large volumes of gold to enable several miners to dig there. This problem is solved by the miners by using a basic Stochastic Diffusion Search.

- At the beginning of the mining procedure, every miner is arbitrarily assigned a hill to mine (his hill hypothesis, *h*).
- On daily basis, each individual miner is assigned with an arbitrarily chosen seam on his hill to mine.
- By end of the day, the possibility that the miner is happy depends upon the quantity of gold mined by him.
- The miners gather by end of every day and the miner who is dissatisfied arbitrarily chooses another miner to talk to. Suppose, the selected miner is satisfied, he gladly informs his peers the identity of the hill in which he is mining (i.e., he shares his hill hypothesis, *h*,). On the other hand, if the selected miner is not happy then he does not share anything while the original miner is again forced to select a fresh hypothesis i.e. to determine the hill that he will be mining the following day – arbitrarily[17].

Among the SDS search, there is a hypothesis, h maintained by every agent to define an achievable solution. In this mining game analogy, agent hypothesis finds a hill. There are two stages that follow this initialization. They are

- Test Phase (e.g. to detect the presence of gold)
- Diffusion Phase (e.g. congregation as well as exchanges of data)

Briefly, while the diffusion phase, an intensification procedure is used around an active agent, namely a good agent whereas a diversification procedure is applied if both agents are inactive. An improved balance among intensification as well as diversification is required for any search method. This balance can be attained through SDS and it terminates once the pre-defined number of function evaluations (FEs) are reached. The suitable solution is enhanced by employing a local search method. The SDS algorithm is as follows:

Initialization of agents and compute their fitness
Repeat
/ Test – phase */*
for each agent, x_i
arbitrarily select an agent, x_j*, from the population*
if $f(x_i) \le f(x_j)$ *then*
$active_i = true$
else
$active_i = false$
endif
endfor
/ Diffusion – phase */*
for each agent, x_i
if $active_i == false$ *then*
arbitrarily select an agent, x_j*, from the population*
if $active_j == true$ *then*
set x_i *to a solution arbitrarily selected from the neighbourhood of* x_j
else
reinitialize x_i *to a solution in an uncharted region*
endif
endif
endfor
Till a terminating condition is fulfilled
Let x_g *be the agent with the best (i.e. minimum) fitness function*
Improve x_g *using a local search method*

Results and Discussion

In this section, the exhaustive search, 3 step search, diamond search, 4 step search, GA based search as well as the proposed SDS methods are evaluated. The table 1 & 2 and figure 2 & 3 shows the computational times and MAD.

Table 1
Computational Times per Pixel

Exhaustive Search	**200.45**
Three Step Search	27.22
Diamond Search	24.32
Four Step Search	22.14
GA based Search	18.76
Proposed SDS	14.32

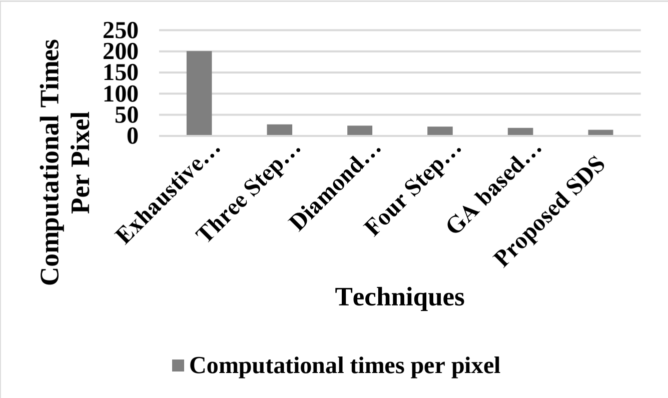

Figure 2: **Computational Times per Pixel**

From the figure 2, it can be observed that the exhaustive search has higher computational times by 152.17% for three step search, by 156.72% for diamond search, by 160.21% for four step search, by 165.76% for GA based search and by 173.32% for proposed SDS.

From the figure 3, it can be observed that the exhaustive search has higher computational times by 6.36% for three step search, by 14.26% for four step search, by 4.82% for GA based search and by 5.67% for proposed SDS. The exhaustive search has lower computational times by 8.54% for diamond search.

Conclusion

In the motion estaimations, search patterns greatly affect the search speed as well as quality of performances. On the basis of motion vectors distribution features of real-world video sequence, this work proposes a novel SDS, a population-based, nature-inspired probabilistic approach, which solves best-match problems mainly by communication between agents. An important feature that makes SDS different from many other optimization techniques is the mathematical framework that proves its convergence to optimal solution even in noisy search spaces. SDS is proposed to investigate dynamically changing environments and in contrast to many connectionist models that find the solution by approaching a specific point in the weight space which results in decreasing of their activity after convergence, SDS is able to continue the exploration over the search space further on after locating the optimum. Results show that the exhaustive search has higher computational times by 152.17% for three step search, by 156.72% for diamond search, by 160.21% for four step search, by 165.76% for GA based search and by 173.32% for proposed SDS.

Table 2
MAD

Exhaustive Search	**12.32**
Three Step Search	11.56
Diamond Search	13.42
Four Step Search	10.68
GA based Search	11.74
Proposed SDS	11.64

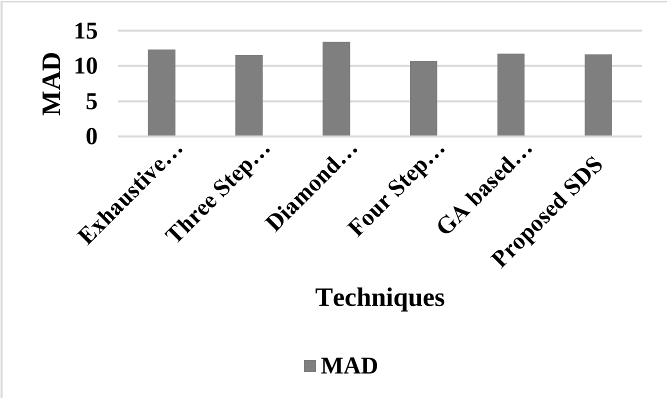

Figure 3: **MAD**

References

1. Sikora T., Digital video coding standards and their role in video communications, *Nato Asi Series of Computer and Systems Sciences*, **174,** 225-252 **(1999)**

2. Liou M., Overview of the p× 64 kbit/s video coding standard, *Communications of the ACM*, **34(4),** 59-63 **(1991)**

3. Sudersanan K. and Ramya R., Design and Implementation of Multi-Standard Video Encoder Supporting Different Coding Standards, *International Journal of Engineering Trends and Technology*, **14(3),** 145-149 **(2014)**

4. Valarmathi S., Vani R. and Sangeetha M., Hardware Software Co-Simulation of Motion Estimation in H, 264 Encoder, In First International Conference on Information Technology Convergence and Services, 263-271 **(2012)**

5. Zuo S., Wang M., Xiao L. and Harbin H., An Efficient VLSI Computation Reduction Scheme in H. 264/AVC Motion Estimation, *WSEAS Transactions on Signal Processing*, **10(1),** 178-187 **(2014)**

6. Colenbrander R.R., Damstra A.S., Korevaar C.W., Verhaar C. A. and Molderink A., Co-design and implementation of the H. 264/AVC motion estimation algorithm using co-simulation, In

Digital System Design Architectures, Methods and Tools, 2008, DSD'08, 11th EUROMICRO Conference, IEEE, 210-215 **(2008)**

7. Krishnan M.E., Gangadharan E. and Kumar N.P., H. 264 Motion Estimation and Applications, INTECH Open Access Publisher, 56-83 **(2012)**

8. Barjatya A., Block matching algorithms for motion estimation, *IEEE Transactions Evolution Computation*, **8(3)**, 225-239 **(2004)**

9. Rahman C.A. and Badawy W., A quarter pel full search block motion estimation architecture for H. 264/AVC, In 2005 IEEE International Conference on Multimedia and Expo, IEEE **(2005)**

10. Pastuszak G., Architecture design of the H. 264/AVC encoder based on rate-distortion optimization, *IEEE Transactions on Circuits and Systems for Video Technology*, **25(11),** 1844-1856 **(2015)**

11. Sun X., Yin B. and Shi Y., A low cost video coding scheme using texture synthesis, In Image and Signal Processing, 2009, CISP'09, 2nd International Congress, IEEE, 1-5 **(2009)**

12. Li G.L., Chen T.Y., Shen M.W., Wen M.H. and Chang T.S., 135-MHz 258-K gates VLSI design for all-intra H. 264/AVC scalable video encoder, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **21(4)**, 636-647 **(2013)**

13. Pastuszak G. and Jakubowski M., Adaptive computationally scalable motion estimation for the hardware H. 264/AVC encoder, *IEEE Transactions on Circuits and Systems for Video Technology*, **23(5)**, 802-812 **(2013)**

14. Liu J., Chen X., Fan Y. and Zeng X., A full-mode FME VLSI architecture based on 8× 8/4× 4 adaptive Hadamard Transform for QFHD H. 264/AVC encoder, In 2011 IEEE/IFIP, 19th International Conference on VLSI and System-on-Chip, IEEE, 434-439 **(2011)**

15. Correa G., Palomino D., Diniz C., Agostini L. and Bampi S., SHBS: A heuristic for fast inter mode decision of H. 264/AVC standard targeting VLSI design, In 2011 IEEE International Conference on Multimedia and Expo, IEEE, 1-4 **(2011)**

16. Yuan X. and Shen X., Block matching algorithm based on particle swarm optimization for motion estimation, In Embedded Software and Systems, 2008, ICESS'08, International Conference, IEEE, 191-195 **(2008)**

17. Al-Rifaie M.M. and Bishop J.M., Stochastic diffusion search review, *Paladyn, Journal of Behavioral Robotics*, **4(3)**, 155-173 **(2013)**.