ENHANCED VOTE COUNT CIRCUIT BASED ON NOR FLASH MEMORY FOR FAST SIMILARITY SEARCH

Haiyan Shu, Wenyu Jiang, Xiaoming Bao, Huan Zhou, and Rongshan Yu

Institute for Infocomm Research, A*STAR, Singapore

ABSTRACT

A memory-based search circuit is introduced in this paper. In this circuit, the conventional memory structure is customized to provide equality comparison for each column of memory array, and a counting circuit is included at each column to record the degree of matches between query and reference data patterns. This customized memory circuit can be used for similarity search applications. Due to its massive parallel processing of equality comparisons and counting operations, the search time using this circuit has $O(1)$ complexity. In addition, it uses NOR flash memory structure and *Enhanced* Vote Count (EVC) interlocked design to achieve low power and high speed. Energy consumption is significantly reduced, by approximately m-fold (m is the number of simultaneously compared pattern bits in EVC), while matching speed is m times faster, compared to original vote count circuit implemented on NOR flash memory structure.

Index Terms— Vote Count, Memory Circuit, EVC Circuit, Similarity Search

1. INTRODUCTION

Similarity search is vital in many application areas. In the context of nearest neighbor similarity search, the query vector is compared with reference vectors, and the closest vectors to the query vector are identified as target items. To obtain these results, the brute force approach is to compute the distance between the query vector and all reference vectors in the dataset. This requires high computation, especially when the feature dimension and the size of dataset are very high.

Vote Count (VC) algorithm [1, 2] is an effective approach to solve the similarity search problem in very high dimension. By using p-stable distribution in the hashing process, high dimension feature vectors are projected to discrete (typically binary) vectors. The similarities between the hashed query vector and reference vectors are compared and the number of hashed dimensions that compare equal is counted. At the heart of the Vote Count and its later improved versions are its customized memory circuits [1, 2, 3], which are also called vote count circuits, where binary pattern matching and counting operations occur *simultaneously* for *each* vector, thus converting complex $O(n)$ time floating-point distance calculations into massively parallel Hamming distance type comparisons taking only $O(1)$ time, i.e., its query time is a constant regardless of the size of the dataset. This makes it an efficient and scalable solution for similarity search.

In particular, a later improved version based on a novel design, which is called *Enhanced* Vote Count (EVC), is a highly energy efficient version of vote count circuit, and its implementation on NAND flash memory structure, referred to as EVC NAND, is introduced in [2, 3]. In EVC, instead of comparing 1 hashing value at a time, m hashing values (which is referred to as a sub-pattern and functionally similar to a hash key) are compared at a time. And the EVC NAND is designed such that a column draws current only if all m values match with the query's, using what is referred to as the *interlocked* design. Power consumption is thus greatly reduced accordingly, by approximately 2^m-fold, while matching speed is m times faster, compared to original vote count circuit implemented on the NAND memory structure.

In this paper, a prototype EVC chip based on NOR flash memory structure is introduced. It implements the same functionality of EVC NAND, but with NOR flash memory structure so as to achieve significantly higher processing speed, with a reasonable trade-off in power consumption. In Section 2, the proposed EVC chip is introduced with exemplary application on multimedia search. The performance of the proposed chip will be detailed in Section 3.1 on the speed and power consumption, and some simulation results on the proposed system are given in Section 3.2. The conclusion is drawn in Section 4.

2. VOTE COUNT CIRCUITS

The basic concept of Vote Count is as follows: first, both reference vectors and the query vector are converted into discrete vectors by some forms of deterministic hashing; next, if the query vector's hashed dimension i is equal to a reference vector's hash dimension i, this reference vector's counter is increased by 1. After L rounds of vote counting, all reference vectors whose vote counter value is at least a specified threshold T are reported as candidate matches and may be further filtered by a more rigorous criterion, e.g. Euclidean distance. This last step of filtering is also referred to as re-ranking.

The Vote Count algorithm, if naively implemented using

software, will have a very high time complexity of $O(n)$. Therefore, specialized hardware architecture, i.e. vote count circuit, has been proposed. In original vote count circuit design, there is a word-line for each row to access memory cell on that row, and there is a bit-line for each column to sense one of the memory cells on that column. An implementation based on DRAM memory structure (referred to as VC DRAM) is given in [1, 2, 3]. The charge and the voltage of the DRAM capacitor depend on its stored value (0 or 1), influence the bit-line (BL) voltage, and thus control the read out during a DRAM read access. When a single cell is being read, the values of all cells in that row on the same *matrix grid* become available simultaneously. In VC DRAM, each column has its own sense-amplifier to sense and read the bit value stored in the cell belonging to that column. After each read, the comparison result may affect the counter for each column. This forms the basic hardware architecture implementing the Vote Count algorithm. This original vote count circuit can be implemented with different memory structures.

Enhanced vote count (EVC) [2, 3] is a highly efficient alternative to the the original vote count circuit, while keeping the essence of vote counting concept. In this design, instead of comparing one value at a time, m values are compared at a time. The number of comparison and counter update operations is reduced significantly by m times when comparing to the original vote count circuit implemented on same memory structure. In Fig. 1, a low power EVC circuit based on the structure of NAND flash memory (referred to as EVC NAND) is given. In EVC NAND, NAND flash is formed by connecting several floating gate transistors (fGTs) in series (together called a NAND string). During reading, all other fGTs are given a *hi* V_G to ensure they conduct, but a *mid* V_G is applied to the fGT of interest (denoted F1). The *hi*'s ensure all other fGTs forming conductive channels, and if F1 is erased, a *mid* V_G is enough to form a conductive channel in F1, and current will flow in the series. If F1 is programmed, *mid* will *not* be enough to form a conductive channel in F1, and the series will *not* conduct. Thus the presence of current (or lackof) decodes charge state of F1 and its respective bit value. With the matrix layout, the *mid* and *hi* are applied to all fGTs on the same row, respectively. And the EVC architecture is designed such that a column draws current only if all m values match with the query's by using what is referred to as the *interlocked* design, where 2 cells are used to represent 1 pattern bit. Power consumption is thus reduced significantly accordingly since only the matched NAND string will draw current and hence consume power.

To further speed up the comparison process, a NOR flash memory structure (Fig. 2) is proposed to implement the EVC functionality, which we refer to as EVC NOR. In this circuit, the *interlocked* design is also adopted, but the exact notation is different from that of EVC NAND. Similar to Fig. 1, each counter will only be increased when a cluster of these m pairs of *interlocked* cells in the same column as the counter are

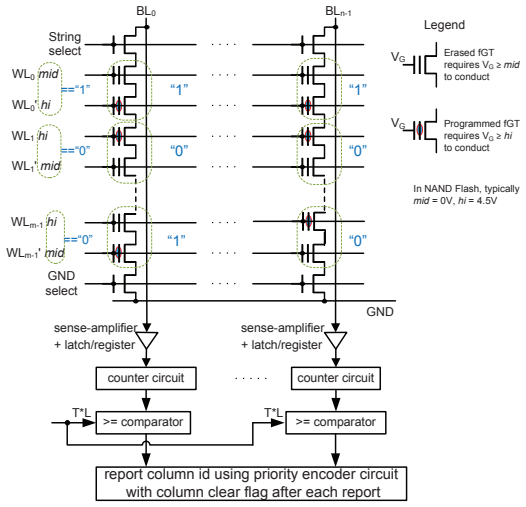


Fig. 1. A NAND flash based EVC implementation. For brevity, only 1 set of NAND string is shown for each data entry (bit-line) in the database. In actual implementation, $L' = \frac{L}{m}$ NAND strings will be connected to each bit-line in a way similar to VC DRAM to support L' comparison and counter update operations.

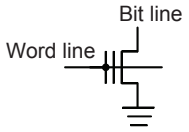


Fig. 2. NOR flash memory.

matched with query data pattern. EVC NOR does not have the property of a column drawing current only when the pattern matches like EVC NAND, rather, its power consumption is similar to that of original vote count circuit implemented on the NOR flash memory structure (referred to as VC NOR). However, because it can compare m values at a time, it can operate m times faster than VC NOR, making its energy per search is m times less. It is also much faster than EVC NAND, with each m-bit comparison taking tens of nanoseconds instead of a few to $tens$ of microseconds.

Based on the proposed NOR flash EVC circuit, a prototype chip (as shown in Fig. 3) has been designed and fabricated in 0.18μm process. In this prototype chip, there are 1024 word-lines and 1024 bit-lines which can accommodate 1024 reference vectors to be retrieved simultaneously. Due to the *interlocked* design where 2 cells are used to represent 1 pattern bit, each feature vector after hashing may have up to 512 dimensions. To minimize the changes to the underlying flash memory circuit, this chip retained the mux that shares each sense-amplifier with 32 columns. Therefore, 32 pattern matching cycles (instead of 1 cycle in a fully optimized version), where each cycle is 60ns, are required to com-

Table 1. Time complexity of original vote count circuit and EVC circuit with different memory structure implementations

	DRAM implementation	NAND implementation	NOR implementation
original vote count circuit	$L \cdot \tau_{par}$	$L \cdot \tau_{ser}$	$L \cdot \tau_{par}$
EVC circuit		$\frac{L}{m} \cdot \tau_{ser}$	$\frac{L}{m} \cdot \tau_{par}$

Fig. 3. Prototype EVC chip based on NOR flash memory structure.

Fig. 4. PCB board.

pletely evaluate one m-bit comparison operation. A test PCB (Fig. 4) is also designed to mount the EVC prototype chip, and integrate with an FPGA development board (specifically, Zynq ZC702) for writing the reference vectors into the flash memory, controlling the comparison operations, reading the comparison results, and communicating with the host PC that is running multimedia search applications.

With this proposed system, similarity search problem can be easily implemented with Vote Count algorithm.

3. PERFORMANCE ANALYSIS

3.1. Speed and power consumption

EVC circuit has a significant speed and power advantage over conventional search systems, especially with large m. This is because EVC's matching is done for m bits simultaneously, leading to $m\times$ speeding up over original vote count circuit which is already much faster than software-based search algorithms. A simple time complexity comparison for vote count circuits with different memory structure implementations is given in Table 1. It is shown that, when the underlying memory structure is the same, EVC circuit presents $m\times$ faster than that of the original vote count circuit. In Table 1, τ_{par} is the DRAM read access time, with typically $\tau_{par} \approx 25 - 50$ns, and τ_{ser} is the NAND flash read access time with typically $\tau_{ser} \approx 5 - 50\mu$s. The read access time of NOR flash is similar to the τ_{par} of DRAM. Generally, the speed of EVC NOR is about 100 to 1000 times faster than that of EVC NAND, making it well suited for fast search applications. In term of power consumption, EVC NOR achieves $m\times$ reduction in energy per search over VC NOR. For EVC NOR, search time is $L/m \cdot \tau_{par}$, hence energy consumption of the matrix per search (E_{search}) is $P_{matrix} \cdot L/m \cdot \tau_{par}$. Our tests on the EVC NOR prototype chip show that $\tau_{par} = 60$ns. Power consumption of the EVC prototype chip P_{matrix} is about 5 to 6mW.

Note that to achieve the same search accuracy, the hashed feature vector length in EVC will generally be larger than that in VC, implying higher memory transistor usage but with the benefit of significantly lower power and energy consumption. Same as the discussion in [3], since the hashed feature vector length is usually no more than a few hundred, it can be seen that, when $n \cdot d$ is around 10000, where n is the size of dataset and d is the dimension of the feature vectors, brute force kNN scheme has similar time complexity to that of the VC NOR and EVC NOR. When $n \cdot d$ increases further, kNN would present increasingly higher time complexity than that of VC NOR and EVC NOR, thus making vote count circuit a more favored solution. For example, in [4], $n = 8\times10^7$ training images ($d = 19$ largest PCA dimensions out of 3072) are used to provide object and scene recognition, implying at least 1.5sec (or 30sec per [4]) to recognize one query image. Although multiple processors can be used to reduce kNN time complexity, it would proportionally add to system cost. In contrast, even with a conservative $L' = 50$ and $\tau_{par} = 60$ns, EVC NOR would only take $\sim 3\,\mu$s. When using the EVC prototype chip, which due to chip design complexity constraints having to take 32 (instead of 1) matching cycles to fully evaluating an m-bit comparison, it will take $\sim 96\,\mu$s, which is still much faster than software based solutions. Furthermore, the high data density of memory chips enables scalability, and is the additional advantage of EVC.

3.2. Simulation result

Based on EVC NOR chip, we have designed and implemented an exemplary similar image search system to evaluate its performance. As introduced in Sec. 2, this chip has 1024 word-lines and 1024 bit-lines. The number of data vectors is limited to 1024. We randomly choose 1024 images from

Table 2. Competitive Analysis with Commercial Software and Open Source Toolkits

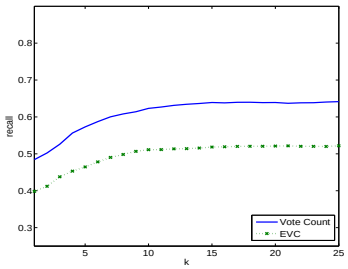
Performance Metrics (1 Million DB items, find top-100 matches)	EVC		Open Source Toolkits
	low-density	high-density	Multi-index Hashing
Search Speed	$0.139ms$	$6.5\mu s$	$3ms$
Energy per Search	$0.75mJ$	$0.03mJ$	$150mJ$
Physical Space	$50cm^2 \cdot 1cm$	$1-2cm^2 \cdot 1cm$	desktop PC

Caltech 101 dataset [5]. GIST descriptors [6] of these images computed at three different scales (8, 8, 4) are used for testing, and the feature dimension is 320 before hashing. kNN search based on the Euclidean distance calculated from GIST vectors is used as ground truth. Considering the inter-locked design, each data vector can have at most 512 dimensions after binary hashing. So, the projection number is set to $L = 512$. The simulation results are presented based on the average behavior over the entire dataset with leave-one-out cross validation. We present the recall of (approximated) kNN search using Hamming distance calculated from VC and EVC on the proposed system for accuracy evaluation.

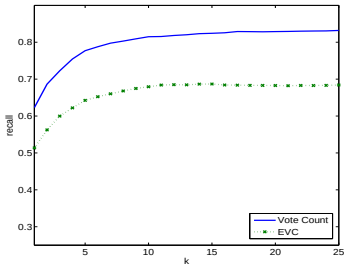
In this chip, 32 (i.e. 16 pairs) memory cells are connected in a sector (similar to a string in NAND Flash). It can be configured as $m = 8$ or 16. Generally, larger value of m means higher computational efficiency (faster speed) and higher power saving. However, the recall may drop with larger m value when the projection number L is the same. In this simulation, we configure $m = 8$ for evaluation. The average recalls of kNN results based on VC and EVC on the proposed system are presented in Fig. 5 (a). It is observed that, EVC performance is not as good as that of VC. Inherently, this is because the projection number L is the same, whereas EVC can only have count up to L/m, i.e., only $L/m + 1$ unique counts or Hamming distances (instead of $L + 1$ in the VC) to differentiate similar items vs. dissimilar items in the dataset. This can be improved with a higher number of projections. And this can also be improved with the concept of weak bit as proposed in [7]. Nevertheless, with m=8, EVC NOR can search 8 times faster and be 8 times more energy efficient than VC NOR.

Next, we gave simulation result for kNN with re-ranking approach. To get k nearest neighbors, $2k$ candidates are retrieved by VC and EVC with proposed system. These candidates are then checked with Euclidean distance to find the top k nearest neighbors. With re-ranking approach, the search accuracies are improved significantly with a limited increase in computational cost. As shown in Fig. 5 (b), both VC and EVC with proposed system present significant improvement over the original approaches.

Last, we compared EVC performance against Multi-Index Hashing (MIH) [8], a state-of-art Hamming distance based kNN algorithm. We ran MIH with a database of 1 million 64-bit hash vectors on a Dell Precision T7500 (Intel Xeon CPU, X5650 @2.67GHz). Despite a $32\times$ slowdown in search speed due to the $32:1$ mux retained in the flash memory



(a) kNN



(b) kNN with re-ranking

Fig. 5. Comparison of kNN performance between VC and EVC with proposed system. L=512, m=8.

circuit, the chip can still search at $>$ 20 times faster than MIH, when extrapolated to a 1000-chip configuration capable of supporting 1 million vectors. This is shown in Table 2, column "EVC low-density". The "EVC high-density" column estimates how a fully optimized EVC NOR chip based system would perform on the same test. Such chip will be based on a more advanced node such as 55nm process, and have the $32:1$ mux removed, so that every column has its own sense-amplifier.

4. CONCLUSION

In this paper, a customized NOR flash memory based EVC chip is introduced. With this chip, paralleled processing for high dimension large dataset search becomes possible. The chip also has low power consumption. We further use the customized chip to implement an exemplary similar image search system and the performance has been presented with promising result.

5. REFERENCES

[1] V. Singh and W. Jiang, "An Algorithm and Hardware Design For Very Fast Similarity Search in High Dimensional Space," *Symposium on Foundations and Practice of Data Mining, part of Conference on IEEE Granular Computing*, Aug 2010.

[2] W. Jiang and R. Yu, "High-Performance, Very Low Power Content-based Search Engine," *ICME 2013 International Workshop on GREEN multimedia: Energy-efficient Multimedia Computing, Communication and Presentation*, Jul. 2013.

[3] Haiyan Shu, Rongshan Yu, Wenyu Jiang, and Wenxian Yang, "Efficient Implementation of k-Nearest Neighbor Classifier Using Vote Count Circuit," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 6, pp. 448 – 452, Jun. 2014.

[4] A. Torralba, R. Fergus, and W. Freeman, "80 million tiny images: a large dataset for non-parametric object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 11, pp. 1958–1970, 2008.

[5] Fei-Fei Li, R. Fergus, and Pietro Perona, "Learning Generative Visual Models From Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)*, 2004.

[6] Aude Oliva and Antonio Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *International Journal of Computer Vision*, vol. 42, pp. 145 –175, 2001.

[7] Haiyan Shu, Wenyu Jiang, and Rongshan Yu, "Study on Weak Bit in Vote Count and its Application in k-Nearest Neighbors Algorithm," *IEEE Int. Conf. Industrial Electronics and Applications*, Jun. 2015.

[8] Mohammad Norouzi, Ali Punjani, and David J. Fleet, "Fast Search in Hamming Space with Multi-Index Hashing," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3108 – 3115, Jun. 2012.