International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) - 2016

Ramanujan Sums Based Image Kernels for Computer Vision

Krishnaprasad P
Department of Computer Engineering
GEC Palakkad
Kerala, India
Email: krishnaprasadpguptha@gmail.com

Ajeesh Ramanujan
Department of Computer Engineering
GEC Palakkad
Kerala, India
Email: ajeeshramanujan@gmail.com

***Abstract*—In the recent history, kernel methods had established themselves as powerful tools for computer vision. In this paper we introduce an integer image kernel function based on Ramanujan Sums which finds its place in image vision. The paper proves the validity of kernel function theoretically and also shows the application of the kernel in image vision. Ramanujan Sums are based on number theory and hence the new kernel matrix will contain only the integer values. Since the image processing involves complex matrix manipulations, the processing based on the new kernel will be computationally effective. The paper shows the applicability of the kernel in various context of image processing. By applying the theory of Ramanujan Sums for image kernel, we will show the intervention of numerical mathematics in machine learning which gives new directions for future research.**

Keywords- Image Kernels; Ramanujan Sums; Edge Detection; Image Gradient; Computer Vision

I. Introduction

Kernel methods have proved successful in all areas of image processing such as optical character recognition, object classification, action recognition, image segmentation, content based image retrieval mainly because of their interpretability and flexibility [3]. By constructing a good kernel function, our aim is to integrate the prior knowledge of humans for a particular problem. Once a promising kernel function has been designed, it can be re-used in any kernel-based algorithm, not just in the context it was originally designed. This gives researchers as well as practitioners a large pool of established kernel functions to choose from, thereby increasing the chances of finding a well-performing one.

Here our interest goes for presenting a novel image kernel based on the theory of Ramanujan Sums. The Ramanujan Sums was introduced by the famous Indian mathematician Srinivasa Ramanujan, in 1918 and it is now known as the Ramanujan Sums $C_q(n)$ [7]. In the last ten years, Ramanujan Sums have aroused some interest in signal processing. There is evidence that these sums can be used to extract periodic components in discrete-time signals. The numerous properties of Ramanujan Sums are the key determining factor of its search for applicability in the area of machine learning. The numerical mathematics have less intervention in machine learning till now, our aim is to prove that, a numerical function can be efficiently used for machine learning by applying Ramanujan Sums as kernel function.

We will establish kernel by applying it in the context edge detection of image experimentally. The entire document is structured as follows: In Section II we will search all the related works and Section III will review the Ramanujan Sums. In Section IV we will introduce the new image kernel and finally we will discuss experimental results and paper will end by conclusion. Through out this paper we will follow the following notations:

(1) The abbreviations lcm and gcd stand for least common multiple and greatest common divisor, respectively.
(2) The notation (k, q) represents the gcd of the integers and k and q. So $(k, q) = 1$ means that, k and q are co-prime.
(3) The quantity $\phi(q)$ is the Euler's totient function. It is equal to the number of integers k in $1 \leq k \leq q$ in satisfying $(k, q) = 1$.
(4) The $\Gamma_q = e^{-j2\pi/q}$ is the q^{th} root of unity, and $\mathbb{C}^q$ the q dimensional space of complex vectors. For a matrix A, transpose, conjugate, and transpose-conjugate are denoted as $A^T, A^*, A^\dagger$ respectively.
(5) For a $1 \times n$ column matrix, M, $cir(M)$ means the matrix obtained by the circular shift of each row. We call this matrix as circulant matrix. The order of the circulant matrix will be $n \times n$.
(6) We use $\diamondsuit\diamondsuit$ to denote the end of theorems, $\triangle\triangle$ will denote end of proof, and $\nabla\nabla$ denote the end of definition.
(7) The $\mathbb{N}$, $\mathbb{I}$ and $\mathbb{R}$ denotes the set of natural numbers, integers and real numbers respectively.

II. Related Works

The relation between the kernel methods and computers vision has been studied widely. Lampart gives an introduction to kernel methods in computer vision from geometric perspective, introducing not only the ubiquitous support vector machines, but also less known techniques for regression, dimensionality reduction, outlier detection and clustering [3]. Zhang et al., studies various non-local kernel regression for image and video restoration tasks [10]. Odone et al., describes methods for building kernels from binary strings for image matching [4].

Ramanujan Sums was introduced by Srinivasa Ramanujan in paper a entitled *"On certain trigonometrical sums and their*

978-1-4673-9939-5/16/$31.00

applications in the theory of numbers" in 1918 [7]. In his paper, he presents a summation formula and studies its relation to number theory in detail. Interesting applications of Ramanujan Sums in the context of signal processing have been examined by a number of authors. Soo-Chang Pei and Kuo-Wei Chang studies a special class of odd-symmetric length-$4N$ periodic signals and they shows the odd Ramanujan Sums are used as weighting coefficients to compute their pure imaginary discrete Fourier transform (DFT) integer-valued coefficients [5]. The traditional methods to get the values of Ramanujan Sums follow the definition and formula in number theory, both of which need factorization information. As it is complex and the amount of time needed is unpredictable in hardware programming, a programmable approach based on the primitive roots of unity is proposed and studied [9]. Planat studies the application of Ramanujan Sums for signal processing of low-frequency noise [6]. K. Grauman et al., presents new fast kernel function which maps unordered feature sets to multi-resolution histograms and computes a weighted histogram intersection [1]. H. Zhang et al., studies image and video restoration via non-local kernel regression [11]. T. Hofmann presents performance evaluation of edge detection techniques for images in a spacial domain which is useful for comparison of methods [3].

The field of kernel methods and image processing is rich enough and research is still going on for finding better results.

III. Review of Ramanujan Sums

In 1918, the famous Indian mathematician Srinivasa Ramanujan introduced a trigonometric summation, now called the Ramanujan Sums $C_q(n)$. It is defined as the sums of the n^{th} powers of the q^{th} primitive roots of the unity. This summation formula has the following form:

$$C_q(n) = \sum_{\substack{k=1 \\ (k,q)=1}}^{q} cos\frac{2\pi kn}{q} \tag{1}$$

or equivalently[8],

$$C_q(n) = \sum_{\substack{k=1 \\ (k,q)=1}}^{q} e^{j2\pi kn/q} \tag{2}$$

The notation (k, q) denotes the greatest common divisor (gcd) of k and q . Its clear that if $(k, q) = 1$, it means that k and q are relatively prime or more precise they co-prime to each other. For example if $q = 5$, co-prime values of k are $k = 1, 2, 3, 4$ and so that

$$C_5(n) = e^{j2\pi n/5} + e^{j4\pi n/5} + e^{j6\pi n/5} + e^{j8\pi n/5}$$

A. Properties of Ramanujan Sums

The q^{th} Ramanujan Sum $(q \geq 1)$ is a sequence n in defined as

$$C_q(n) = \sum_{\substack{k=1 \\ (k,q)=1}}^{q} e^{j2\pi kn/q} = \sum_{\substack{k=1 \\ (k,q)=1}}^{q} \Gamma_q^{-kn} \tag{3}$$

Thus the summation runs over only those values of k that are coprime to q. So, the sum (3) has precisely $\phi(q)$ terms, and

$$C_q(0) = \phi(q) \tag{4}$$

Also, we can show that Ramanujan Sums are periodic functions with period q.
i.e.,

$$C_q(n + q) = C_q(n) \tag{5}$$

Equation (3) also shows that the DFT of $C_q(n)$ is:

$$C_q[k] = \begin{cases} q, \textit{if } (k, q) = 1 \\ 0, otherwise \end{cases} \tag{6}$$

If $(k, q) = 1$, then it follows that $(q - k, q) = 1$ as well. Since $\Gamma_q^{-(q-k)} = \Gamma_q^k = (\Gamma_q^k)^*$, Then it then follows that the summation in (2) is real valued.

$$C_q(n) = \sum_{\substack{k=1 \\ (k,q)=1}}^{q} cos\frac{2\pi kn}{q} \tag{7}$$

From (7) it is clear that $C_q(n)$ is symmetric, periodic real sequence. By symmetric we mean that, $C_q(n) = C_q(-n)$. The periodic nature can be shown as

$$C_q(n) = C_q(q - n) \tag{8}$$

which is consistent with the fact that the DFT $C_q[k]$ is real. Thus any Ramanujan Sum $C_q(n)$ is a real, symmetric, and periodic sequence in n . Here are the first few Ramanujan sequences, shown for one period $0 \leq n \leq q - 1$.

$C_1(n) = 1$
$C_2(n) = 1, -1$
$C_3(n) = 2, -1, -1$
$C_4(n) = 2, 0, -2, 0$
$C_5(n) = 4, -1, -1, -1, -1$

IV. Image Kernel

An image kernel is a small matrix used to apply effects like the ones you might find in Photoshop or Gimp, such as blurring, sharpening, outlining or embossing. They're also used in machine learning for 'feature extraction', a technique for determining the most important portions of an image. We present a novel kernel matrix based on Ramanujan Sums for computer vision. Before introducing the kernel matrix lets prove the validity of the new kernel. In general kernel can be defined as:

Definition: Let f be a symmetric function defined on non-empty set χ such that $f : \chi \times \chi \to \mathbb{R}$, then f is a positive definite kernel on χ if:

$$\sum_{i,j=1}^{n} c_i c_j k(x_i, x_j) \geq 0 \tag{9}$$

holds for any $n \in \mathbb{N}, x_1, \ldots, x_n \in \chi$ and $c_1, \ldots, c_n \in \mathbb{R}$.
$\nabla\nabla$

From definition one can easily deduce that the set of positive definite kernels is a closed, convex pointed cone. The positive definiteness of kernel functions translates in practice into the positive definiteness of so called Gram matrices,(also called Kernel matrices). The kernel matrix can be built on a sample of points $X = \{x_i\}_{i \in \mathbb{I}}$ in χ as,

$$K_X = [k(x_i, x_j)] \tag{10}$$

where $i, j \in \mathbb{I}$

A. Ramanujan Kernel Function

The Ramanujan kernel function can be defined as

$$\sum_{\substack{k=1 \\ (k,q)=1}}^{q} e^{j2\pi kn/q} \tag{11}$$

Each element of kernel matrix corresponding the kernel function can be obtained by

$$\sum_{\substack{k=1 \\ (k,q)=1}}^{q} e^{j2\pi ki/q} \tag{12}$$

where i ($0 \leq i \leq q$) in the equation denotes the i^{th}position in the matrix.

The entire matrix is calculated from :

$$cir(\sum_{\substack{k=1 \\ (k,q)=1}}^{q} e^{j2\pi ki/q}) \tag{13}$$

Thus the Kernel matrix will be of the order of $q \times q$ The general form of a $q \times q$ kernel matrix can be shown as bellow:

$$M_q = \begin{bmatrix} C_q(0) & C_q(1) & C_q(2) & \dots & C_q(q-1) \\ C_q(q-1) & C_q(0) & C_q(1) & \dots & C_q(q-2) \\ \vdots & \vdots & \vdots & & \vdots \\ C_q(1) & C_q(2) & C_q(3) & \dots & C_q(0) \end{bmatrix}$$

Kernel functions establish the characteristics of SVM model and level of non linearity. A necessary and sufficient condition for a simple inner product kernel to be valid is that it must satisfy Mercer's theorem.

B. Mercer's Theorem

A symmetric function $K(x, y)$ can be expressed as an inner product

$$K(x, y) =< \phi(x), \phi(y) >$$

for some ϕ if and only if $K(x, y)$ is positive semidefinite. or, equivalently:

$$M = \begin{bmatrix} K(x_1, x_1) & K(x_1, x_2) & \dots \\ K(x_1, x_2) & K(x_2, x_2) & \dots \\ \vdots & & \end{bmatrix}$$

is positive semi definite for any collection $x_1 ... x_n$.

$\diamond\diamond$

C. Validity of Ramanujan Kernel

To prove, Ramanujan kernel is a valid kernel function, it is enough to prove the kernel matrix is positive semidefinite. Diagonalizing the circulant matrix M_q by DFT matrix as

$$M_q = A^{-1} \Lambda_q A = \frac{A^\dagger \Lambda_q A}{q} \tag{14}$$

Here $C_q[k]$ are the DFT coefficients of the Ramanujan sequence $C_q(n)$, i.e.,

$$C_q[k] = \sum_{n=0}^{q-1} C_q(n) A^{nk} = \begin{cases} q, \mathit{if}\ (k, q) = 1 \\ 0, \mathit{otherwise} \end{cases} \tag{15}$$

Since $\frac{A}{\sqrt{q}}$ is unitary, we see that (14) is equivalent to

$$M_q A^\dagger = A^\dagger \Lambda_q \tag{16}$$

By conjugating both sides and using the fact that $C_q(n)$ and $C_q[k]$ are real and $A = A^T$ we get

$$M_q A = A \Lambda_q \tag{17}$$

So the columns of A are the eigenvectors of M_q with corresponding eigenvalues $C_q[k]$. It means M_q is Hermitian with non-negative eigenvalues $\in \{0, q\}$, it is positive semidefinite.

$\triangle\triangle$

V. Experiments and Results

Let's demonstrate the working of the kernel first. For $q = 2$, kernel matrix is given by:

$$M_q = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

For a given input image, we represent the image as matrix. The matrix contains numbers, between 0 and 255, which each correspond to the brightness of one pixel in a picture of a flower.

Consider the input image:

Fig. 1. Input Image

Figure 1 can be converted into matrix of values between 0 to 255 as in Figure 2:

```
2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02
2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02
2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02 2.460000000000000000e+02
2.460000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02
2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02
2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02 2.470000000000000000e+02
2.470000000000000000e+02 2.470000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02
2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02
2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02
2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02
2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02 2.490000000000000000e+02
2.490000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02
2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02
2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02
2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02
2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02 2.500000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02 2.510000000000000000e+02
2.510000000000000000e+02 2.510000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02
2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.530000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02 2.540000000000000000e+02
```

Fig. 2. Matrix Output

For each 3×3 block of pixels in the image on Figure 2, we multiply each pixel by the corresponding entry of the kernel and then take the sum. That sum becomes a new pixel in the image. For example,

$$\begin{pmatrix} \underset{\times 2}{162} + \underset{\times -1}{173} + \underset{\times -1}{173} \\ + \underset{\times -1}{172} + \underset{\times 2}{178} + \underset{\times -1}{179} \\ + \underset{\times -1}{174} + \underset{\times -1}{178} + \underset{\times 2}{179} \end{pmatrix} = \boxed{-11}$$

A. Image Gradient Using Ramanujan Kernel

In case of image processing find the gradient has its own position. Given a blurred image, we can also produce the gradients for an image. For a surface such as an array of pixels, the gradient is the direction in which the color changes the fastest. We will use filters to implement the image gradients. We will actually produce two gradients, the x-gradient (telling how fast the color changes along each row), and the y-gradient (telling how fast the color changes along each column). Here are the kernels that are used.
x-gradient filter kernel:

$$\begin{bmatrix} -1 & 1 \end{bmatrix}$$

y-gradient filter kernel:

$$\begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

Using these kernels gives us two separate images, one of which tells us how fast the color changes (at each pixel) in the x-direction, and one telling us how fast it changes in the y-direction. Finally, we use this information to determine the direction of the maximum rate of change in color, and we orient our brush strokes perpendicular to that direction. These kernels are frequently used in popular applications.

We have the kernel matrix corresponding to the $q = 1$ as

$$M_q = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

It means that Ramanujan image kernel can be used as gradient matrix.

B. Edge Detection Using RS Kernel

The experiment is performed for edge detection of given image. We will use 3^{rd} order kernel matrix. We will convert the input image into matrix of values between 0 and 255 and then apply the Ramanujan image kernel matrix of order three. The result of the experiment after applying the kernel is shown in Figure 3:

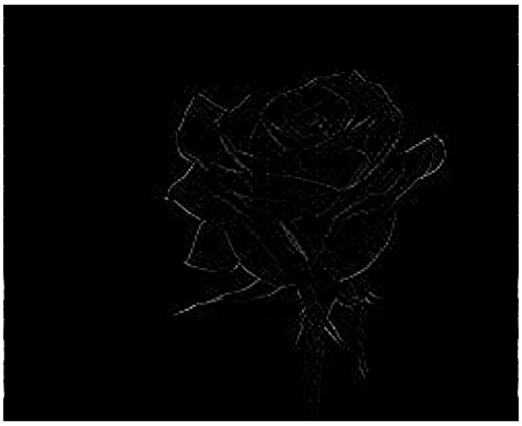

Fig. 3. Resulting Image

The result shows that the3^{rd} order matrix performs well for edge detection.

VI. Conclusion and Future Works

The search for powerful image kernels never stops because of the powerful nature of kernel trick. The image kernel based on the Ramanujan Sums adds one more kernel to the pool of image kernels. Since the new kernel is entirely based on the numerical computations, the computational effectiveness of the new kernel can not be questioned. The results of 3^{rd} order kernel matrix for edge detection is promising. Since the Ramanujan Sums based kernel relates theory of machine learning with numerical mathematics, the unexplored powers of the kernel should be mined.

REFERENCES

[1] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," Proceedings of the IEEE International Conference on Computer Vision, vol. 2, no. 1, pp. 1716–1732, January 2005.

[2] T. Hofmann, B. Scholkopf and Alexander J. Smola, "Kernel methods in machine learning," Institute of Mathematical Statistics, The Annals of Statistics, vol. 36, no. 3, pp. 1171–1220, 2008.

[3] C. H. Lampart, "Kernel methods in computer vision," Journal of Foundations and Trends in Computer Graphics and Vision vol. 3, no. 20, pp. 193-285, 2009.

[4] F. Odone, A. Barla, and A. Verri, "Building kernels from binary strings for image matching," IEEE transactions on image processing, vol. 14, pp. 169 - 180, February 2005.

[5] S. Pei and Kuo-Wei Chang, "Odd ramanujan sums of complex roots of unity," IEEE Transactions on Signal Processing, vol. 14, no. 1, pp. 1716–1732, January 2007.

[6] M. Planat, "Ramanujan Sums for Signal Processing of Low Frequency Noise," Frequency Control Symposium and PDA Exhibition, IEEE International, pp. 715 - 720, 2002.

[7] S. Ramanujan, "On certain trigonometrical sums and their applications in the theory of numbers," Transactions of the Cambridge Philosophical Society, vol. 24, No.13, pp. 259-276, 1918

[8] P. P. Vaidyanathan, "Ramanujan Sums in the Context of Signal Processing Part I: Fundamentals," IEEE transactions on signal processing, vol. 62, no. 16, pp. 4145 - 4157, August 2014.

[9] G. Xujing Z. Lina, S. Jiadong and W. Zulin ,"A programmable approach to evaluate Ramanujan Sums," Proceedings of ICCSEE, January 2013.

[10] H. Zhang, J. Yang, Y. Zhang and Thomas S. Huang, "Non-local kernel regression for image and video restoration," Proceedings of the 11th European conference on computer vision, pp. 566-579 , 2010.

[11] H. Zhang, J. Yang, Y. Zhang and Thomas S. Huang, "Image and video restoration via non-local kernel regression," IEEE transactions Cybernetics, vol. 43, pp. 1035 - 1046, June 2013.