

Use of Costas arrays in subpixel metrology

John J. Healy^a, Gavin Sweeney^b, David Mas^c and John T. Sheridan^b

^aDepartment of Electronic Engineering, National University of Ireland Maynooth, Maynooth, Co. Kildare, Ireland;

^bSchool of Electrical, Electronic and Communications Engineering, University College Dublin, Belfield Dublin 4, Ireland;

^cDepartment of Optics, Univ. Alicante P.O. Box, 99, 03080 Alicante Spain

ABSTRACT

Subpixel methods increase the accuracy and efficiency of image detectors, processing units, and algorithms and provide very cost-effective systems for object tracking. A recently proposed method permits micropixel and submicropixel accuracies providing certain design constraints on the target are met. In this paper, we explore the use of Costas arrays - permutation matrices with ideal auto-ambiguity properties - for the design of such targets.

Keywords: Superresolution, metrology, motion tracking, Costas array

1. INTRODUCTION

Motion tracking has a wide range of applications, and so receives considerable attention in the literature. Optical techniques provide a logical approach to motion tracking, being non-contact, highly-sensitive and potentially low cost. Optical motion tracking essentially consists of measurement and interpretation of differences between sequential images, or equivalently two frames of video. Depending on the nature of the target, we may need to detect a single pixel difference in order to determine the movement of the target. Design of a simple optical pattern to place on the target allows us to manipulate the form of the information received by the camera. Consequently, we can detect sub-pixel motion, or other motions traditionally difficult to extract. In this paper, we present some initial results from the exploration the use of sparse 2D arrays called Costas arrays for target design to simplify sub-pixel motion tracking.

It was recently shown that the accuracy of object tracking systems can be theoretically enhanced by several orders of magnitude,¹ exceeding the three orders of magnitude found in the literature.²⁻⁴ Using numerical simulations, it was demonstrated that micropixel resolutions can be achieved, and that further improvements are possible provided that the sensor has sufficient pixels.

Consider a digital camera with $N \times N$ pixels in an imaging system. Each pixel has coordinates (i, j) , with $i, j \in 1, 2, \dots, N$. We can consider a binary detection (i.e. the pixel gives an 'on/off' response, depending on whether or not it detects light above some small threshold) or entertain the possibility of distinguishing between grey-scale levels. We wish to design a binary intensity target which has some larger number of pixels $nN \times nN$, and is imaged onto the camera face. We neglect the effects of the point spread function in this discussion. We cannot determine the exact position of any one single image spot on the camera, since based on the detected signal all that is known is that it falls inside the active area of a single pixel. Movement of the spot within this pixel area is undetectable. However, movement of a spot from one pixel to the next is visible if we can distinguish greyscale levels. If we are using binary detection, we can only detect a pixel change providing the pixel the spot exited is now empty, or the pixel it has entered is empty now. Motion of an individual spot may therefore be tracked only some of the time, but if we design the relative location of the spots appropriately, the camera may register unique patterns for each sub-pixel shift ($1/n$ th of a pixel) of the target.

Further author information: (Send correspondence to J.T.S.)

J.J.H.: Email: jhealy@eeng.nuim.ie, Telephone: +353 1 708 6383

J.T.S.: E-mail: john.sheridan@ucd.ie, Telephone: +353 1 716 1927

D.M.: E-mail: david.mas@ua.es

In the method presented in this paper, subpixel accuracy may only be obtained when tracking moving targets. The target is defined so that the dots in the target produce spots at the camera that are smaller than the pixel area. This defines an over-sampled array that expands the Fourier limits. According to discrete Fourier theory, spatial interpolation enlarges the bandwidth by zero padding the spectrum. We must emphasize that no information is added and the extra bandwidth remains empty. We are making use of the temporal dimension, and higher accuracy is obtained provided that there is a temporal change in the target. Detection is produced in the temporal domain, but the particular form of the target permits us to use the time variations to increase our information about the spatial shift. In this sense, the method performs a spatial multiplexing of the information, and the temporal variation is stored in the new frequency band so that a subpixel accuracy improvement is obtained.

The remainder of this paper is structured as follows. In Section 2, we introduce Costas arrays. This is followed by a discussion of some preliminary results in Section 2.3. Finally, we present our conclusions.

2. COSTAS ARRAYS

We propose a solution to the problem posed in the introduction involving permutation matrices with ideal auto-correlation properties, known as *Costas arrays* after their discoverer, John Costas. Costas permutations were discovered in the 1960s as a time-frequency description of optimal wave fields for RADAR and SONAR systems,^{5,6} in the sense that they prescribed frequency-hopping patterns with optimal noise robustness. The difficulty of their construction was quickly recognized, and in the 1980s the first (and thus far last) mathematical methods for their construction, based on the theory of finite fields, were presented.^{7,8} Along with those found by the “brute force” method of exhaustive search of all permutations of a particular length (currently all orders $\leq 29^9$), the Costas arrays constructed using these methods and minor variations thereof are practically the only known arrays. Orders 32 and 33 are the smallest ones for which no example of a Costas permutation is yet known.

The auto-ambiguity properties of Costas arrays have led to their proposal for used in numerous applications. In addition, the arrays are the object of theoretical mathematical interest. It should be noted that Costas arrays have not been studied out of the wider context of positioning problems in discrete mathematics: connections have been found with non-attacking Kings, Queens,⁸ and Rooks¹⁰ in chess, Vatican and Florentine arrays,¹¹ Tuscan arrays,^{12,13} and so forth, through which important contributions have been made to Costas arrays themselves. Furthermore, it seems likely that these arrays may provide alternative solutions to the problem discussed in this paper.

2.1 Definition

Let $n \in \mathbb{N}$; we introduce the notation $[n] = \{1, \dots, n\}$, and we denote, as usual, by a permutation of order n any bijection on $[n]$.

Definition Let $p : [n] \rightarrow [n]$ be a permutation of order $n \in \mathbb{N}$. p has the Costas property iff

$$\forall i, j, k \in [n] : i + k, j + k \in [n], \quad p(i + k) - p(i) = p(j + k) - p(j) \Rightarrow i = j$$

We will also need the concept of the *difference triangle* of a permutation:

Definition Let p be a permutation of order n ; its difference triangle $T(p)$ is the collection of vectors $\{t_i(p) : i \in [n - 1]\}$, where $t_i(p) = (t_{ij}(p) = p(i + j) - p(j) : j = 1, \dots, n - i)$. The vector $t_i(p)$ is the i th row of $T(p)$.

The difference triangle is aptly named because it actually looks like a triangle when its rows are written one under the other and centered; its elements take values in the set $\{-(n - 1), \dots, n - 1\} - \{0\}$, and it is easy to see that it contains exactly $n - i$ absolute values equal to i in it, $i \in [n - 1]$, as the absolute difference between any two integers in $\{1, \dots, n\}$ is taken exactly once. It is also easy to see that the Costas property has the equivalent formulation that no row of the difference triangle contains two equal entries.

The difference triangle for the Costas permutation $\{2, 5, 1, 6, 4, 3\}$ (of order 6) is:

$$\begin{array}{cccccc}
2 & 5 & 1 & 6 & 4 & 3 \\
3 & -4 & 5 & -2 & -1 & \\
-1 & 1 & 3 & -3 & & \\
& 4 & -1 & 2 & & \\
& & 2 & -2 & & \\
& & & 1 & &
\end{array}$$

Permutations correspond bijectively to permutation arrays, namely arrays whose elements are either 0 or 1, so that there is exactly one element equal to 1 per row and column. It is customary to represent 1s by dots and 0s by blanks in the graphical depiction of such an array. There are many alternatives in choosing the permutation array that corresponds to a particular permutation, and a convention must be followed:

Definition Let p be a permutation of order $n \in \mathbb{N}$; the $n \times n$ permutation array $A^p = [a_{ij}^p]$ corresponding to p is constructed by setting $a_{n+1-p(i),i} = 1, i \in [n]$, and all other elements equal to 0.

Note that, according to this convention, the $p(i)$ th element of column i , starting to count *from the bottom up*, is set to 1, $i \in [n]$. Some authors flip this convention vertically, which is more consistent with matrix theory.

The array representation has the advantage of making some results obvious: e.g., we can see immediately that the Costas property has the equivalent formulation that no four dots form a parallelogram and that for no three dots does one lie at the midpoint of the other two. As these relative dot positions remain unchanged under rotation, horizontal and vertical flip, and transposition of the array, we see that one Costas array leads to the construction of eight in total, or four in case it is symmetric.

We normally do not distinguish between permutations and permutation arrays and use the terms interchangeably: we will prefer below to talk about permutations rather than arrays, unless we want to specifically emphasize array aspects and properties.

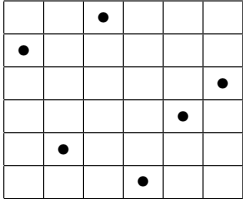
A permutation matrix is an $N \times N$ matrix with one unit-valued entry per row, and one per column, all other entries being zero-valued. Two common alternative representations are used,

1. A permutation, $f[n]$, where $0 \leq n \leq N - 1$ indexes the columns, and $f[n]$ indicates the row in the n th column which contains a 1.
2. An $N \times N$ grid, with a \bullet indicating the location of each 1.

For example, the following matrix is the Costas array $\{2, 5, 1, 6, 4, 3\}$,

$$\begin{pmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0
\end{pmatrix}$$

Equivalently, we can represent this array as,



or as the permutation $\{2, 5, 1, 6, 4, 3\}$. In this paper, as in the literature, these representations will be used completely interchangeably.

Figure 1 shows the autocorrelation of a Costas array. Note that the strong central peak, while the side-lobes are no larger than one unit. One may picture the autocorrelation operation as follows: a pair of cards have holes punched in them in the locations of the unit entries of the Costas array. These are laid on top of one another. When they exactly align, all N holes pass light. However, if we move the top card (vertically and horizontally - no rotation is allowed) so that any other pair of holes align, these are the only pair of holes which align. This is a consequence of the requirement that each vector joining pairs of elements of the permutation array be unique. This ‘thumbtack’ autocorrelation of the array is characteristic of Costas arrays, and explains its robustness to noise.

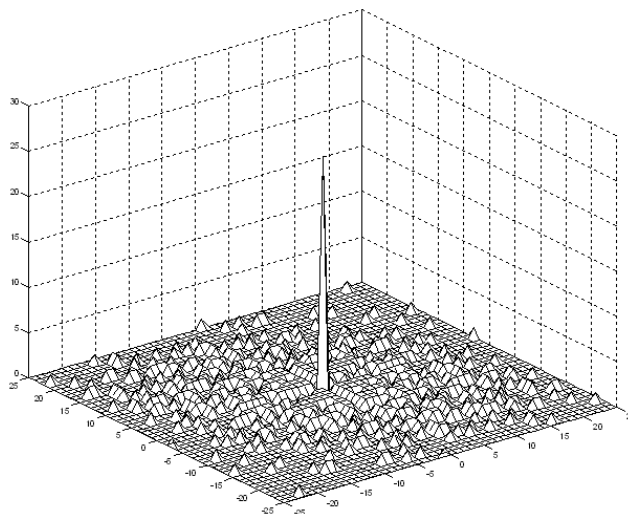


Figure 1. The autocorrelation of a Costas array.

2.2 Enumeration of Costas Arrays

One of the first Costas arrays papers⁷ noted that the question of whether or not Costas arrays exist for all orders remained unresolved. This remains the case at the time of writing. Early efforts to find mathematical generation techniques yielded two methods. Subsequent efforts have found only minor extensions of the known techniques. Many arrays are known for which cannot be generated by known mathematical techniques. These are termed ‘sporadic arrays’.

We will now introduce the main mathematical generation methods for Costas arrays. We will then discuss brute force computer searching.

2.2.1 Algebraic Construction Methods

Costas assumed that such arrays were well studied in combinatorics, and contacted Solomon Golomb, a highly respected scholar in that field. Golomb worked on the problem and also sent it to some colleagues, resulting in the only known mathematical generation techniques to date.

The arrays generated with these techniques often have properties distinct from other Costas arrays. For this reason, and because their definitions are more tractable for analysis than the Costas condition, the different types of algebraic construction are often analyzed separately.

Both of the major generation techniques use finite field theory; each one has several special cases and minor variants. As such, the techniques only work for certain orders near primes and powers of primes.

2.2.2 Algebraic Construction Methods: Welch arrays

THEOREM 2.1 (WELCH CONSTRUCTION $W_1(p, \alpha, c)$). *Let p be a prime, let α be a primitive root of the finite field $\mathbb{F}(p)$ of p elements, and let $c \in [p - 1] - 1$ be a constant; then, the function $f : [p - 1] \rightarrow [p - 1]$ where $f(i) = \alpha^{i-1+c} \bmod p$ is a bijection with the Costas property. The reason for the presence of -1 in the exponent is that, when $c = 0$, 1 is a fixed point: $f(1) = 1$. We refer to arrays generated with $c \neq 0$ as circular shifts of the array generated by $c = 0$ for the same p and α .*

Welch arrays are generated for fields of prime size, resulting in an array of size one less than the prime. Given a field over a prime, p , and a primitive element of that prime, α , the numbers $\alpha^1, \alpha^2, \alpha^3, \dots, \alpha^{p-1}$ modulo p generate a permutation. The permutation array this indexes is a Costas array. This is proven in.⁷

Welch Costas arrays have *anti-reflective symmetry* (see below, and also⁷):

Definition Let $f : [2n] \rightarrow [2n]$; f has anti-reflective symmetry iff

$$f(n + i) + f(i) = 2n + 1, \quad i \in [n],$$

which means, in other words, that the right half of A^f is the vertical flip of the left half.

2.2.3 Algebraic Construction Methods: Lempel-Golomb arrays

THEOREM 2.2 (GOLOMB CONSTRUCTION $G_2(p, m, a, b)$). *Let p be a prime, $m \in \mathbb{N}$, and let α, β be primitive roots of the finite field $\mathbb{F}(p^m)$ of $q = p^m$ elements; then, the function $f : [q - 2] \rightarrow [q - 2]$ where $\alpha^i + \beta^{f(i)} = 1$ is a bijection with the Costas property.*

If $\alpha = \beta$, the array is known as a *Lempel Costas array*, and is symmetric about the main diagonal.

Lempel-Golomb arrays are generated for fields of prime or power of prime size. The resulting arrays are of size two less than the prime or power of prime.

2.2.4 Exhaustive Computer Search

Exhaustive computer search consists of checking each of the $N!$ permutation arrays of size N for the Costas property. This has been undertaken for $N \leq 29$. Drakakis claims that this search is a problem of order $N^3 N!$,¹⁴ though this measure does not take account of the optimizations used by current researchers.

Every Costas array of size $n \leq 29$ has been found by exhaustive computer search:⁹ they add up to about 150000 arrays in total. The number of arrays for a given order as enumerated in this fashion is given in the following table:

N	$C(N)$	N	$C(N)$	N	$C(N)$	N	$C(N)$	N	$C(N)$
1	1	7	200	13	12852	19	10240	25	88
2	2	8	444	14	17252	20	6464	26	56
3	4	9	760	15	19612	21	3536	27	204
4	12	10	2160	16	21104	22	2052	28	712
5	40	11	4368	17	18276	23	872	29	164
6	116	12	7852	18	15096	24	200		

Fig. 2(a) plots the data of Table 2.2.4. The peak value occurs at $N = 16$. Note that the value at $N = 27$ breaks the monotonic decrease for $16 \leq N < 27$. Fig. 2(b) is a log plot of the density of the arrays in the search space.

The most recent brute force searches have involved supercomputing resources equivalent to hundreds of years of a single PC's computation, and each increase in size requires a five-fold increase in resources. It seems unlikely that these results will be extended much further in the near future. However, for our purposes in this paper, algebraically generated methods are sufficient for our task.

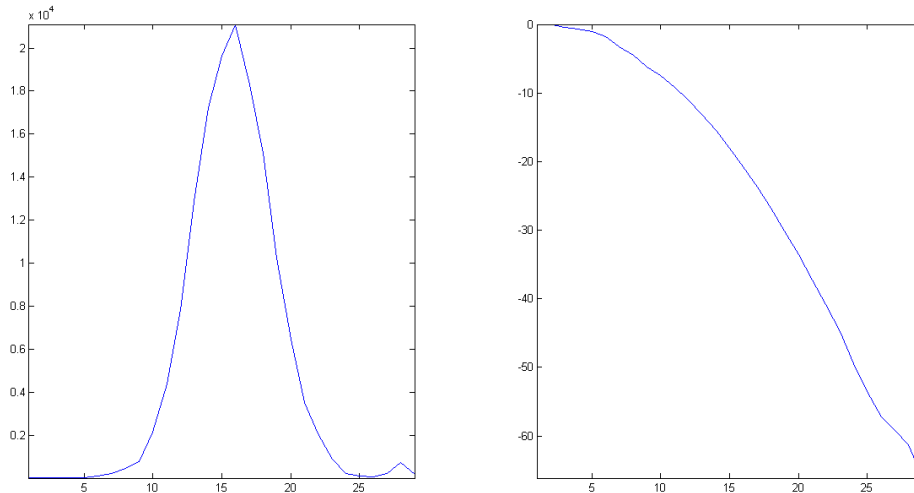


Figure 2. (a) The number of Costas arrays plotted against array size. (b) The logarithm of the number of Costas arrays divided by the number of permutations plotted against a given array size.

2.3 Some early results

In this Section we consider some Costas array based designs for targets and examine their properties. Throughout, we will consider only 16×16 targets imaged onto CCD arrays of 4×4 pixels. Consider first a motionless target that has been imaged carefully so as to align the target perfectly with the CCD face. The value recorded by each pixel will be the number of spots in the corresponding 4×4 group of ‘micropixels’ of the target. This value will be an integer between 0 and 4. We can adjust the sensitivity of the camera and/or the intensity of the target illumination to scale the returned value to take maximal advantage of the dynamic range of the camera. We wish to measure sub-pixel motion. We therefore move the target up to 3 micropixels to the left or right and up or down. Each pixel of the CCD now corresponds with a different group of 4×4 micropixels to before, and so may return a different greyscale value. In each of the examples in this Section, we show all 49 possible CCD patterns resulting from the various sub-pixel shifts of the example target. We require that each of these patterns be unique in order for us to identify the sub-pixel motion. We may place further constraints on the uniqueness at a later point. We will also consider whether the patterns are unique if we binary threshold the arrays, i.e. the CCD pixel returns a zero if there no spots in the corresponding 4×4 group of micropixels of the target, and saturates otherwise.

First, for illustrative purposes, let us consider a chirp array. This is not a Costas array, and has poor autocorrelation properties. The array is given by $v = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$. The resulting pattern is shown in Fig. 3. Note that the result is symmetrical about the diagonal and the anti-diagonal. The patterns are not unique: for example, the $(-2,0)$, $(-1,1)$ and $(0,2)$ shifts all produce the same pattern. The situation only worsens if we consider binary thresholding.

Next, we considered a Golomb array. The Golomb construction method doesn’t produce any 16×16 arrays, so we use an array generated by a small variation, whereby a 17×17 array which exhibits a corner dot is reduced to size. This produces just one unique array (up to the dihedral symmetry group), given by $v = \{6, 14, 10, 7, 9, 1, 4, 16, 5, 3, 12, 11, 15, 2, 13, 8\}$. The results are show in Fig. 4. The patterns are unique on an intensity level. Furthermore, they are also unique on a binary level. It remains to be confirmed whether this occurs in general for the Golomb construction.

Next, we considered a series of arrays generated using the Welch method. We were able to generated three unique arrays, each of which exhibited the same properties: they are unique on an intensity level, but some duplications arise when we binary threshold the resulting patterns. One of the sets of patterns, arising from the

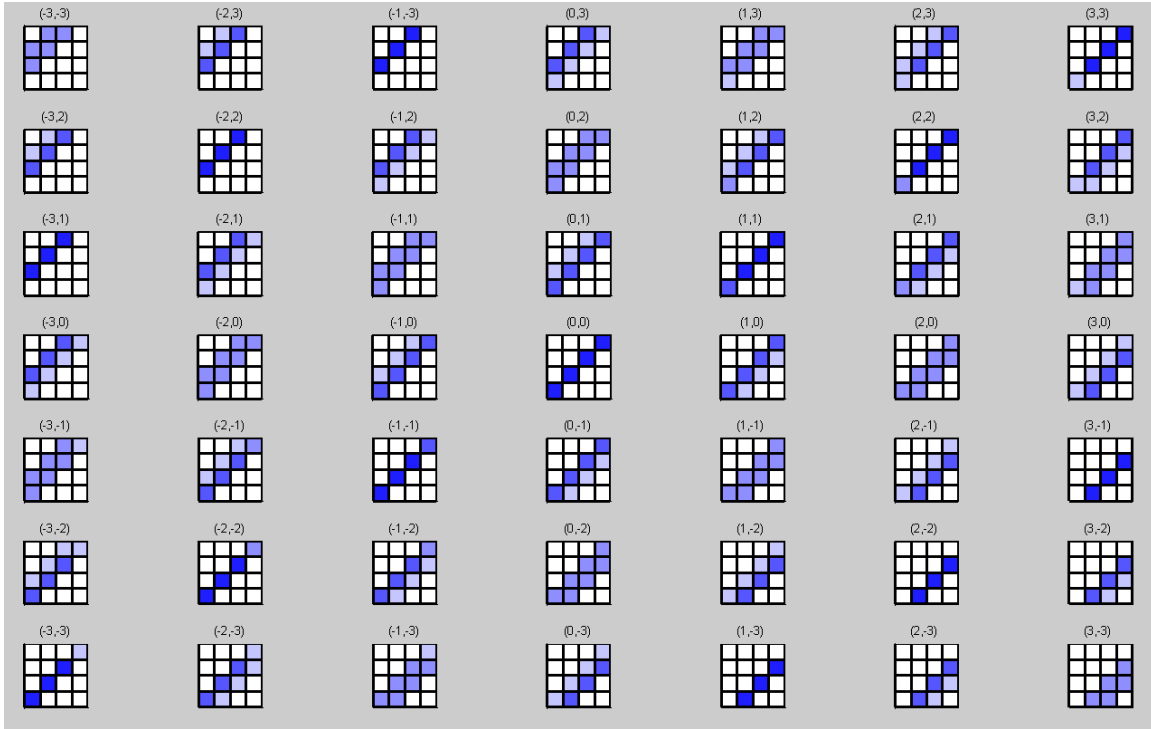


Figure 3. Results from diagonal array, a chirp. Not the replication of patterns. As a result, we cannot distinguish certain sub-pixel motions from one another.

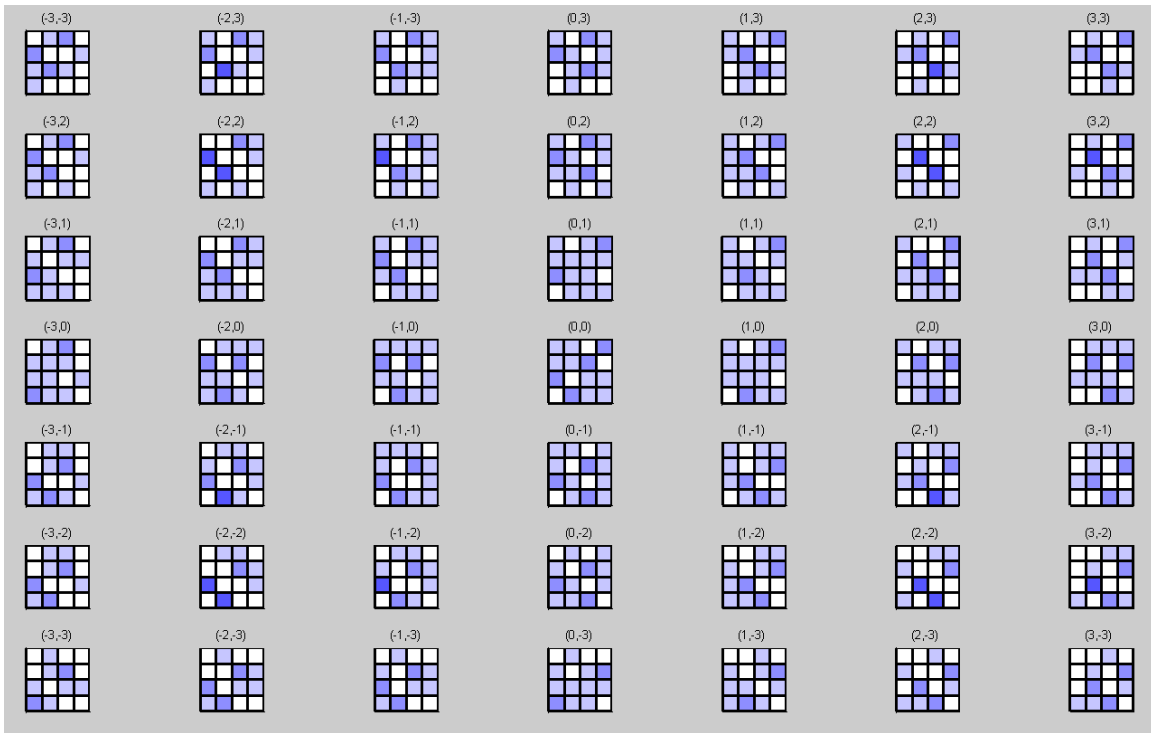


Figure 4. CCD patterns arising from a Golomb Costas array based target. Note that the patterns are unique, even if we take all 'on' values to be the same.

pattern $v = \{4, 14, 15, 8, 11, 9, 13, 2, 10, 5, 1, 3, 16, 7, 6, 12\}$, is provided in Fig. 5. This makes the Welch arrays very interesting for further study.

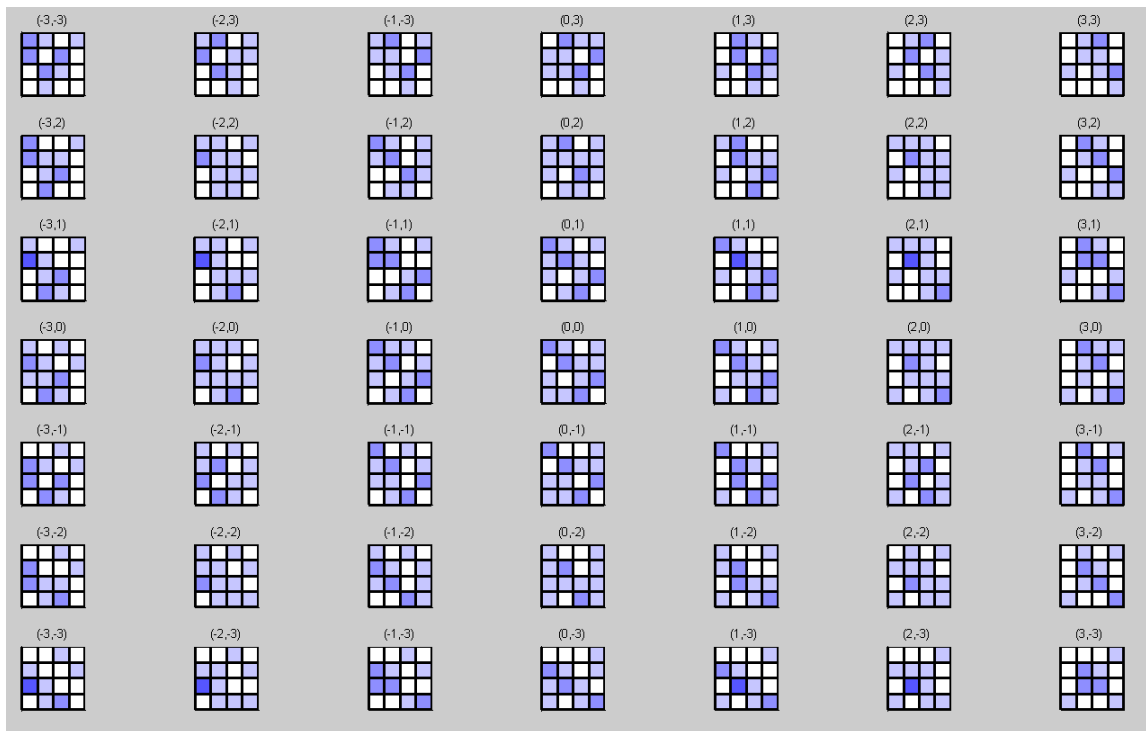


Figure 5. There are more Welch arrays at this size, so we have been able to examine a number of them. Each one was, like this, somewhat promising: generating a unique set of patterns for all sub-pixel shifts. However, binary thresholding does not work in this case.

Finally, in Fig. 6, we considered a sporadically arising Costas array, $v = \{11, 13, 16, 14, 5, 2, 15, 10, 6, 7, 4, 1, 12, 9, 3, 8\}$. This exhibits no duplication at any level. However, it is unlikely we will be able to draw any general conclusions about sporadic arrays.

3. CONCLUSION

In this paper, we have reported some initial results arising from the investigation of the use of Costas arrays to create targets for sub-pixel motion detection. The most promising techniques are based on the algebraic methods of generating Costas arrays.

Future work will likely involve leveraging the periodicity properties of the arrays to make periodic targets, and so avoid the need for any alignment. We also intend an investigation into the correlation properties of the resulting CCD patterns for more noise-robust detection, and expect further development of targets arising from this study. Experimental validation also remains to be carried out.

ACKNOWLEDGMENTS

JJH acknowledges the support of the National University of Ireland (NUI) Post-doctoral Fellowship in the Sciences. DM acknowledges the support of the Spanish Ministerio de Economía y Competitividad through the project BIA2011-22704, the Generalitat Valenciana through the project PROMETEO/2011/021, and the University of Alicante through the project GRE10-09. The authors also would like to acknowledge the support of Science Foundation Ireland and Enterprise Ireland under the National Development Plan.

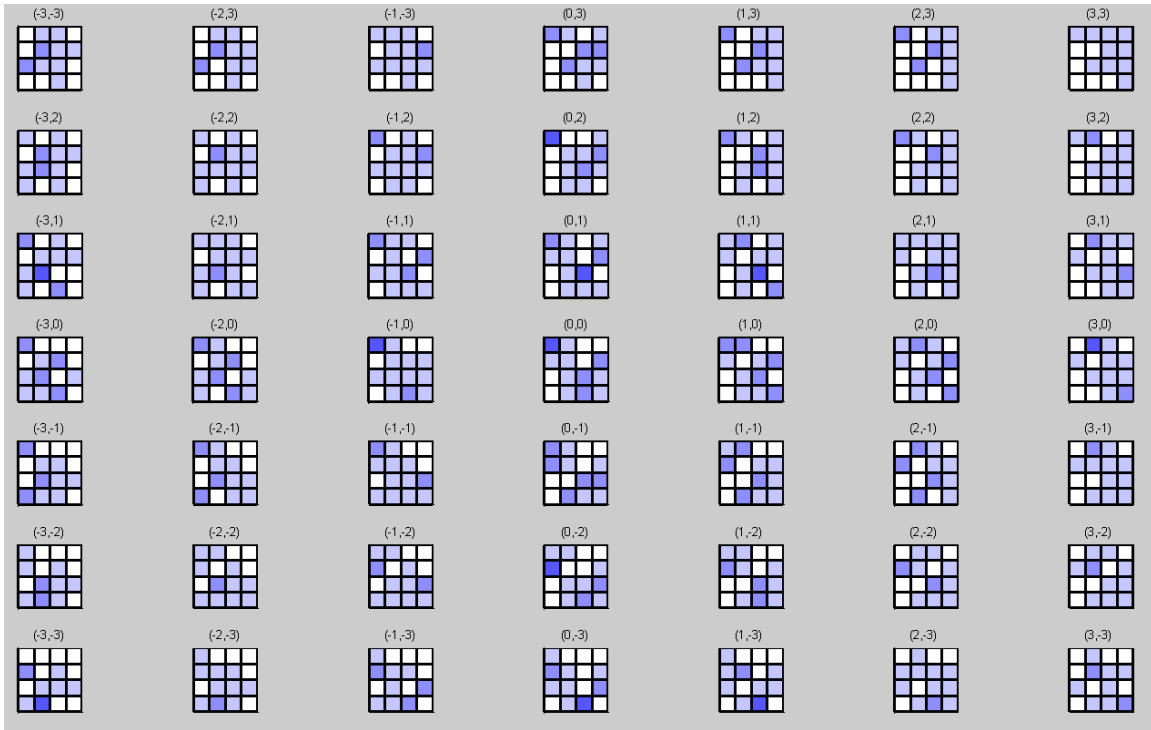


Figure 6. Sporadic array. The patterns are unique, but it is difficult to establish general results for sporadic arrays.

REFERENCES

- [1] Mas, D., Ferrer, B., Sheridan, J. T., and Espinosa, J., "Resolution limits to object tracking with subpixel accuracy," *Opt. Lett.* **37**, 4877–4879 (December 2012).
- [2] Lee, S. H., Baday, M., Tjioe, M., Simonson, P. D., Zhang, R., Cai, E., and Selvin, P. R., "Using fixed fiduciary markers for stage drift correction," *Opt. Expr.* **20**(11), 12177–12183 (2012).
- [3] Mas, D., Espinosa, J., Roig, A. B., Ferrer, B., Perez, J., and Illueca, C., "Measurement of wide frequency range structural microvibrations with a pocket digital camera and sub-pixel techniques," *Appl. Opt.* **51**(14), 2664–2671 (2012).
- [4] Kurita, R. and Weeks, E. R., "Experimental study of random-close-packed colloidal particles," *Phys. Rev. E* **82**, 011403 (Jul 2010).
- [5] Costas, J. P., "Medium constraints on sonar design and performance," Tech. Rep. Class 1 Rep. R65EMH33, GE Co. (1965). A synopsis of this report appeared in the Eascon. Conv. Rec., 1975, pp. 68A–68L.
- [6] Costas, J. P., "A study of detection waveforms having nearly ideal range-doppler ambiguity properties," *Proc. IEEE* **72**(8), 996–1009 (1984).
- [7] Golomb, S. W., "Algebraic constructions for Costas arrays," *J. Combinatorial Th. Series A* **37**(1), 13–21 (1984).
- [8] Golomb, S. W. and Taylor, H., "Constructions and properties of Costas arrays," *Proc. IEEE* **72**(9), 1143–1163 (1984).
- [9] Drakakis, K., Iorio, F., Rickard, S., and Walsh, J., "Results of the enumeration of costas arrays of order 29," *Advances in Mathematics of Communications*.
- [10] Taylor, H., "Non-attacking rooks with distinct differences," tech. rep., EE Systems, University of Southern California (1984).
- [11] Song, H. and Golomb, S., "Generalized Welch-Costas sequences and their application to Vatican arrays," in [*Finite fields: theory, applications and algorithms. 2nd international conference*], 341–352 (1994).
- [12] Etzion, T., Golomb, S. W., and Taylor, H., "Tuscan-k squares," *Adv. Appl. Math.* **10**(2), 164–174 (1989).

- [13] Golomb, S. W., Etzion, T., and Taylor, H., "Polygonal path constructions for tuscan-k squares," *Ars Combinatoria* **30**, 97–140 (1990).
- [14] Drakakis, K., "A review of Costas arrays," *Journal of Applied Mathematics* **2006**, 1–32 (2006).