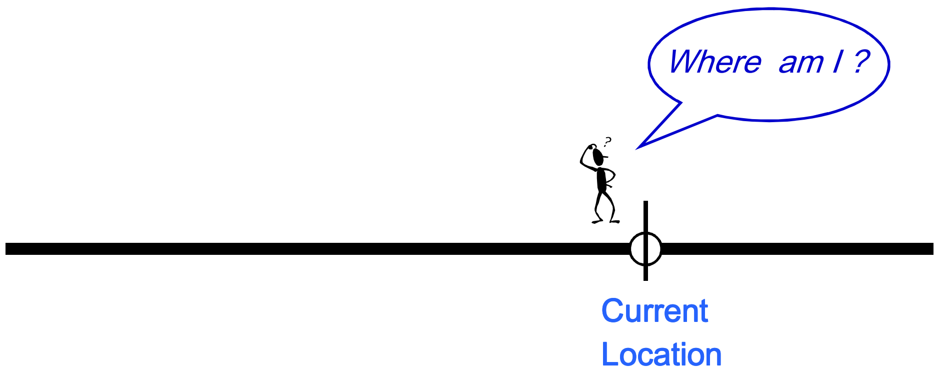
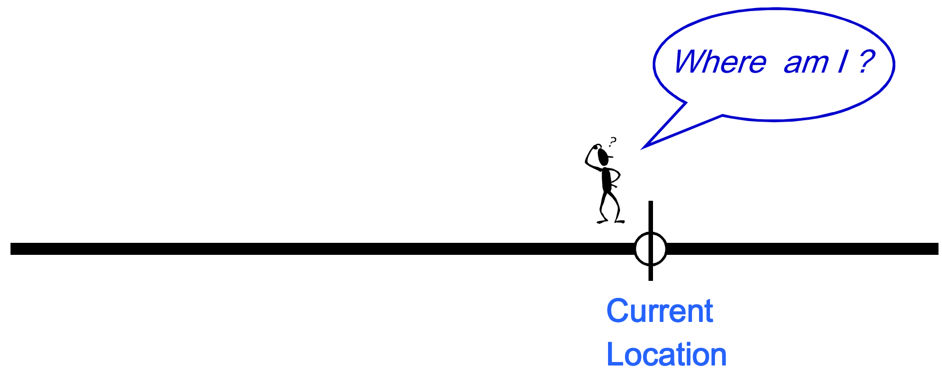
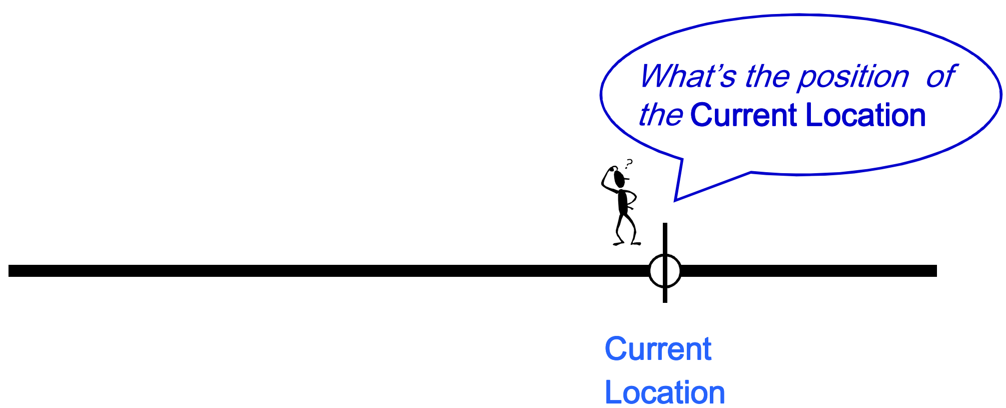
Pseudorandom Coding Techniques for Position Measurement

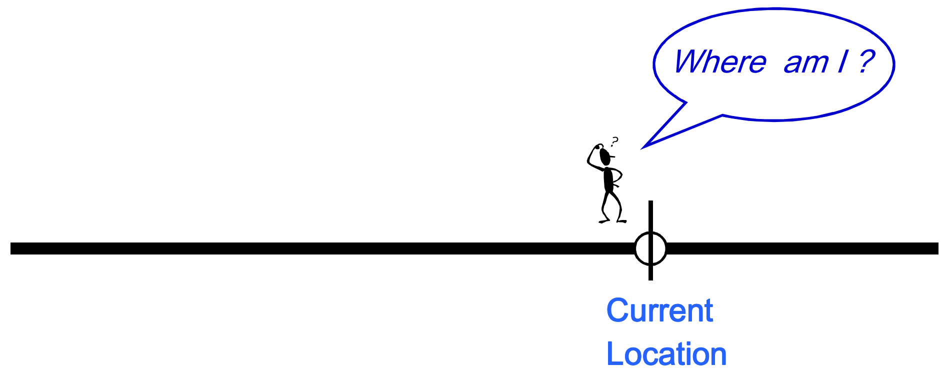
Emil M. Petriu, Dr. Eng., FIEEE
School of Electrical Engineering and Computer Science
University of Ottawa, Canada
http://www.site.uottawa.ca/~petriu

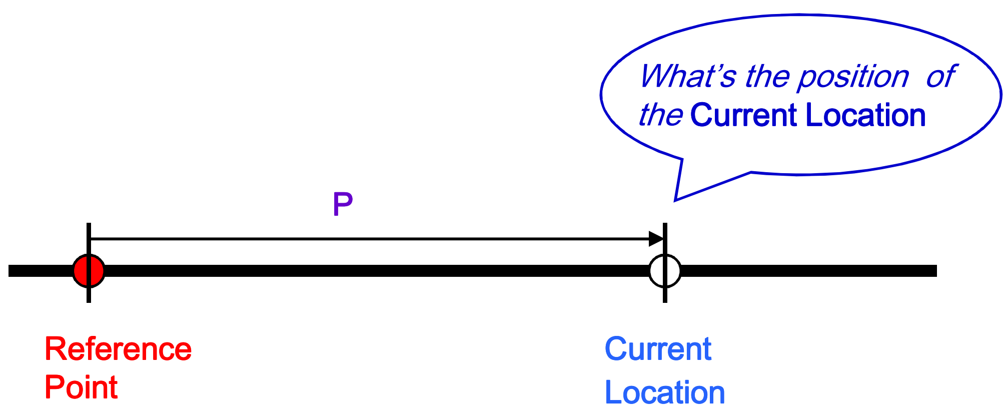

Where am I ?


Where am I ?
Current
Location

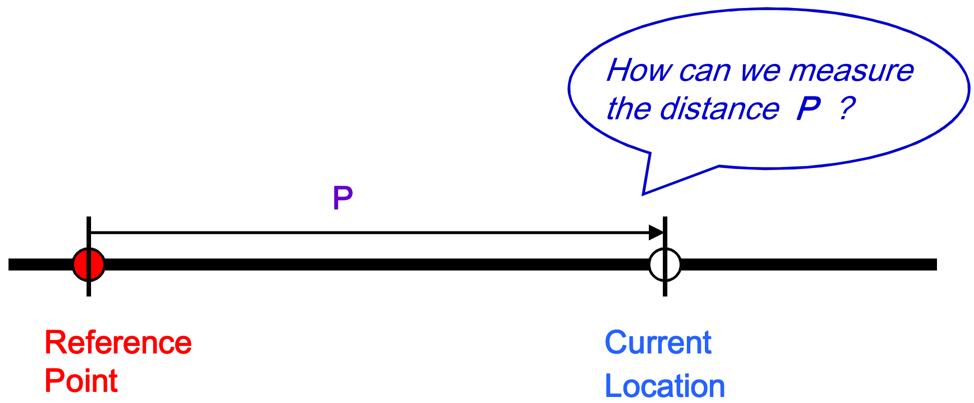

Where am I ?
Current
Location

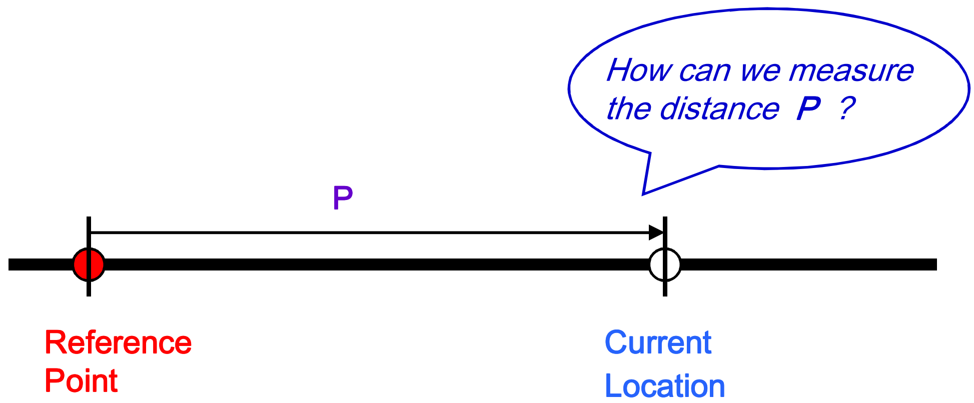

What's the position of
the Current Location
Current
Location



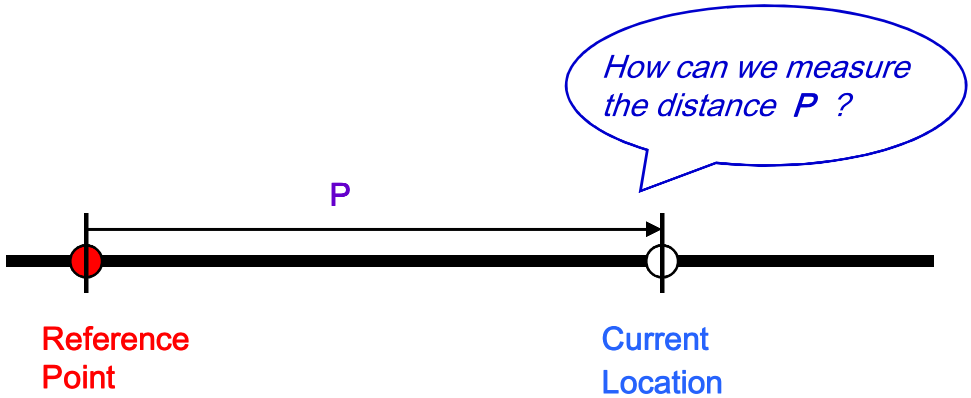


The position of the **Current Location** is defined by its distance **P** from a given **Reference Point**


How can we measure
the distance P ?
P
Reference
Point
Current
Location



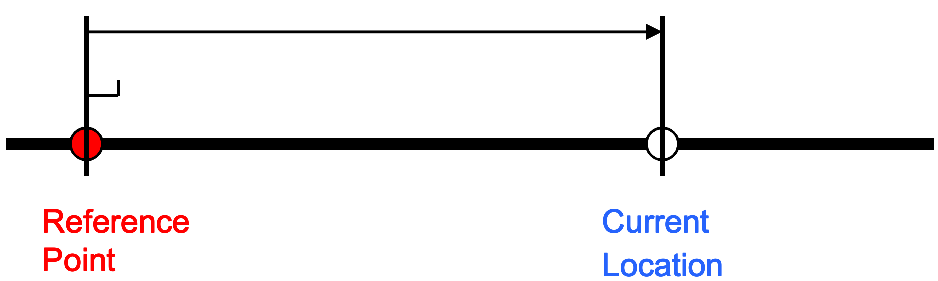
The distance P can be measured by finding out how many units of length q are between the Current Location and the Reference Point .

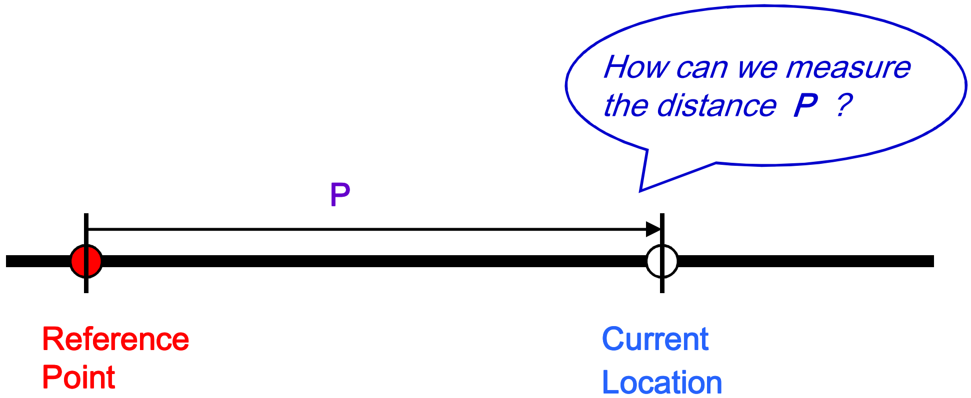


The distance P can be measured by finding out how many units of length q are between the Current Location and the Reference Point .

One way of doing this is to incrementally count the number p of units q that are needed to cover the distance P.

$$P = \sum_{n=0}^{1} n \cdot q$$

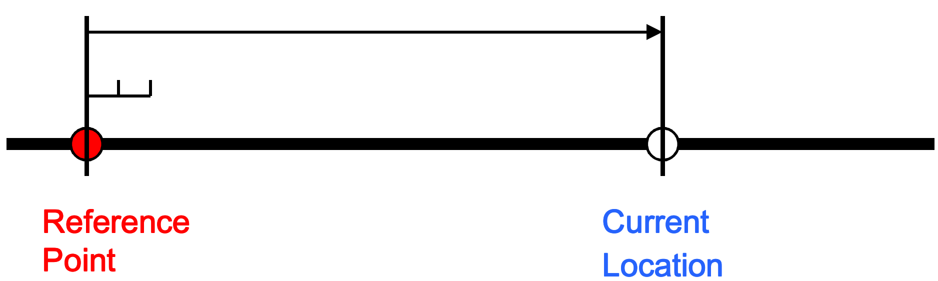


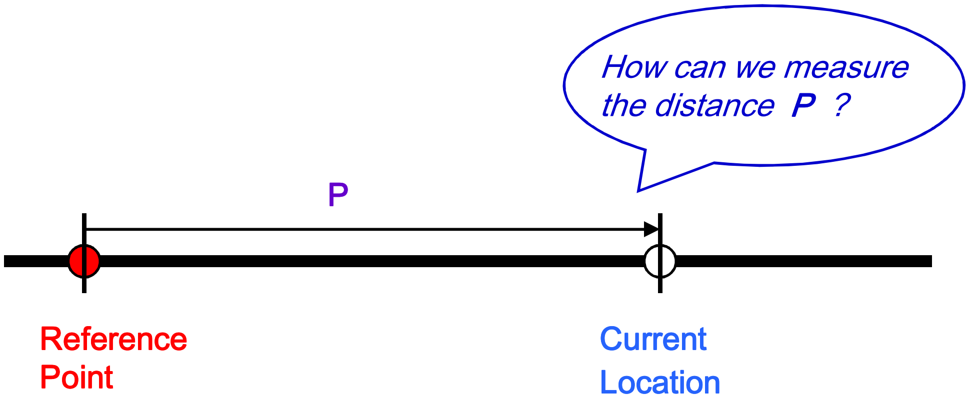


The distance P can be measured by finding out how many units of length q are between the Current Location and the Reference Point .

One way of doing this is to incrementally count the number p of units q that are needed to cover the distance P.

$$P = \sum_{n=0}^{2} n \cdot q$$

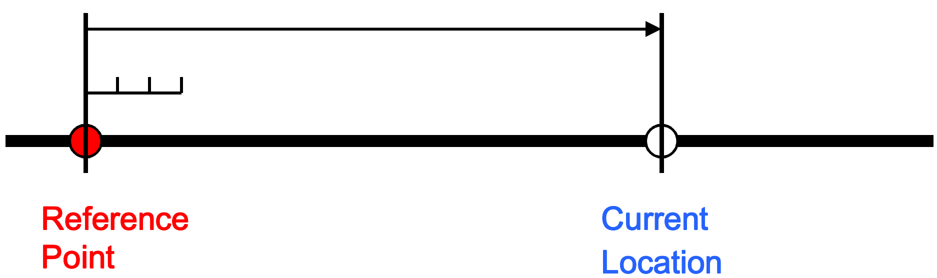


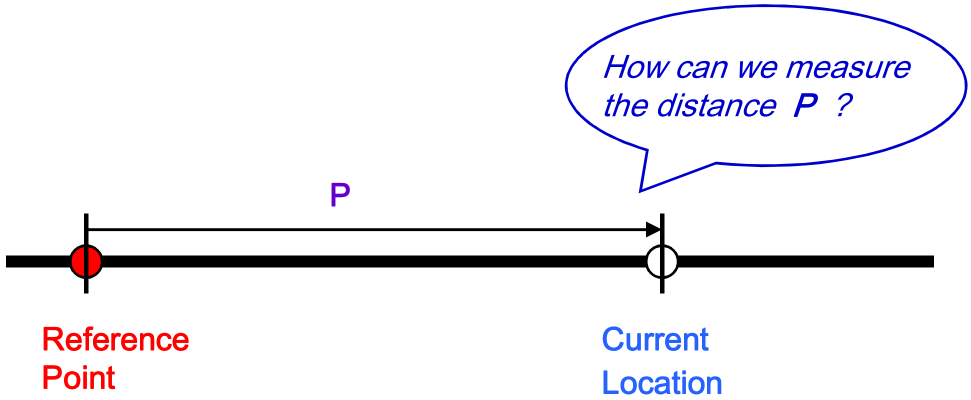


The distance **P** can be measured by finding out how many units of length **q** are between the **Current Location** and the **Reference Point .**

One way of doing this is to incrementally count the number **p** of units **q** that are needed to cover the distance **P**.

$$P = \sum_{n=0}^{3} n \cdot q$$

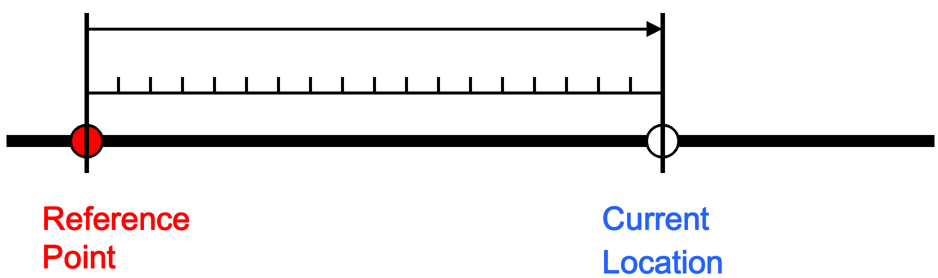


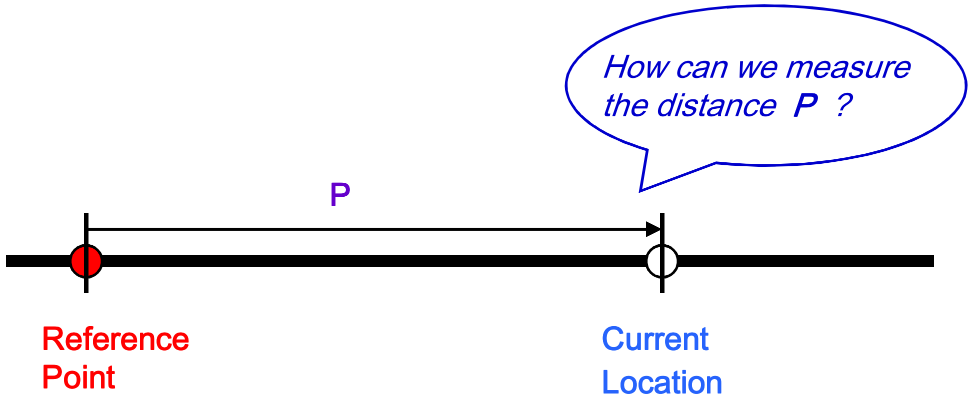


The distance P can be measured by finding out how many units of length q are between the Current Location and the Reference Point .

One way of doing this is to incrementally count the number p of units q that are needed to cover the distance P.

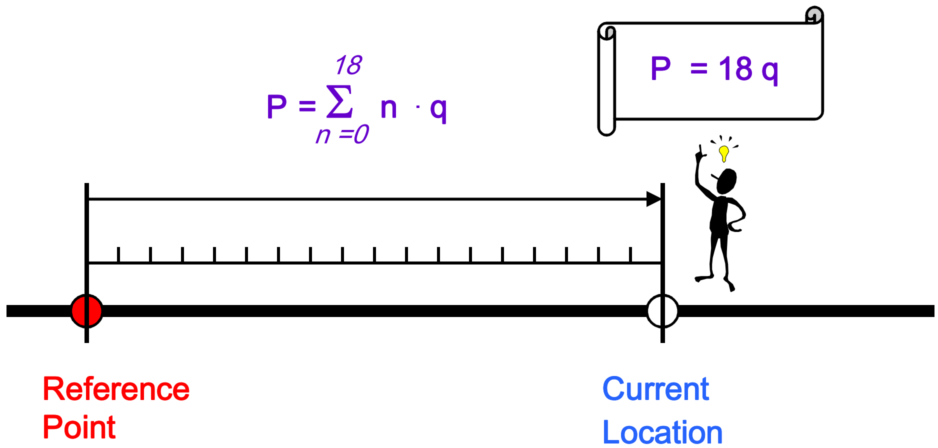
$$P = \sum_{n=0}^{18} n \cdot q$$





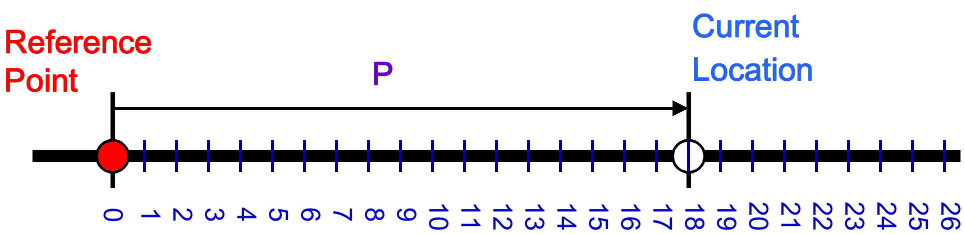
The distance **P** can be measured by finding out how many units of length **q** are between the **Current Location** and the **Reference Point .**

One way of doing this is to incrementally count the number **p** of units **q** that are needed to cover the distance **P**.

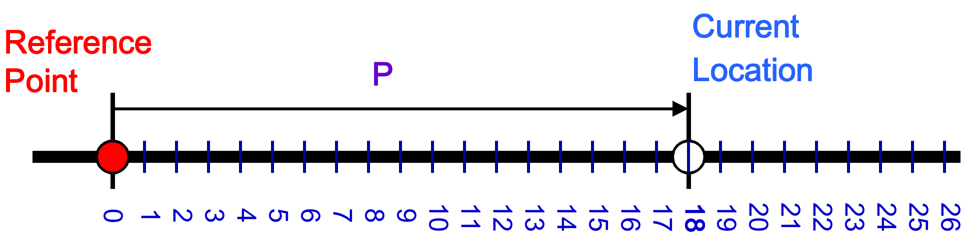


INCREMENTAL MEASUREMENT

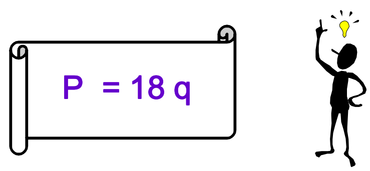
Another way of measuring the position **P** of the **Current Location** relative to the **Reference Point** is to *a priori* mark the number of units **q** corresponding to each position.



Another way of measuring the position **P** of the **Current Location** relative to the **Reference Point** is to *a priori* mark the number of units **q** corresponding to each position.



…. the position **P** is recovered instantaneously by reading the code marked in front of the **Current Location**



☆ ABSOLUTE MEASUREMENT

MEASURING POSITION

Encoders are digital position transducers which are the most convenient for computer interfacing.

Incremental encoders are relative-position transducers which generate a number of pulses proportional with the traveled rotation angle. They are less expensive and offer a higher resolution than the absolute encoders. As a disadvantage, incremental encoders have to be initialized by moving them in a reference (“zero”) position when power is restored after an outage.

Absolute encoders are attractive for joint control applications because their position is recovered immediately and they do not accumulate errors as incremental encoders may do. Absolute encoders have a distinct n-bit code (natural binary, Gray, BCD) marked on each quantization interval of a rotating scale. The absolute position is recovered by reading the specific code written on the quantization interval currently facing the reference marker.

The metrological performance of a position recovery system three parameters: **accuracy**, **repeatability**, and **resolution.**

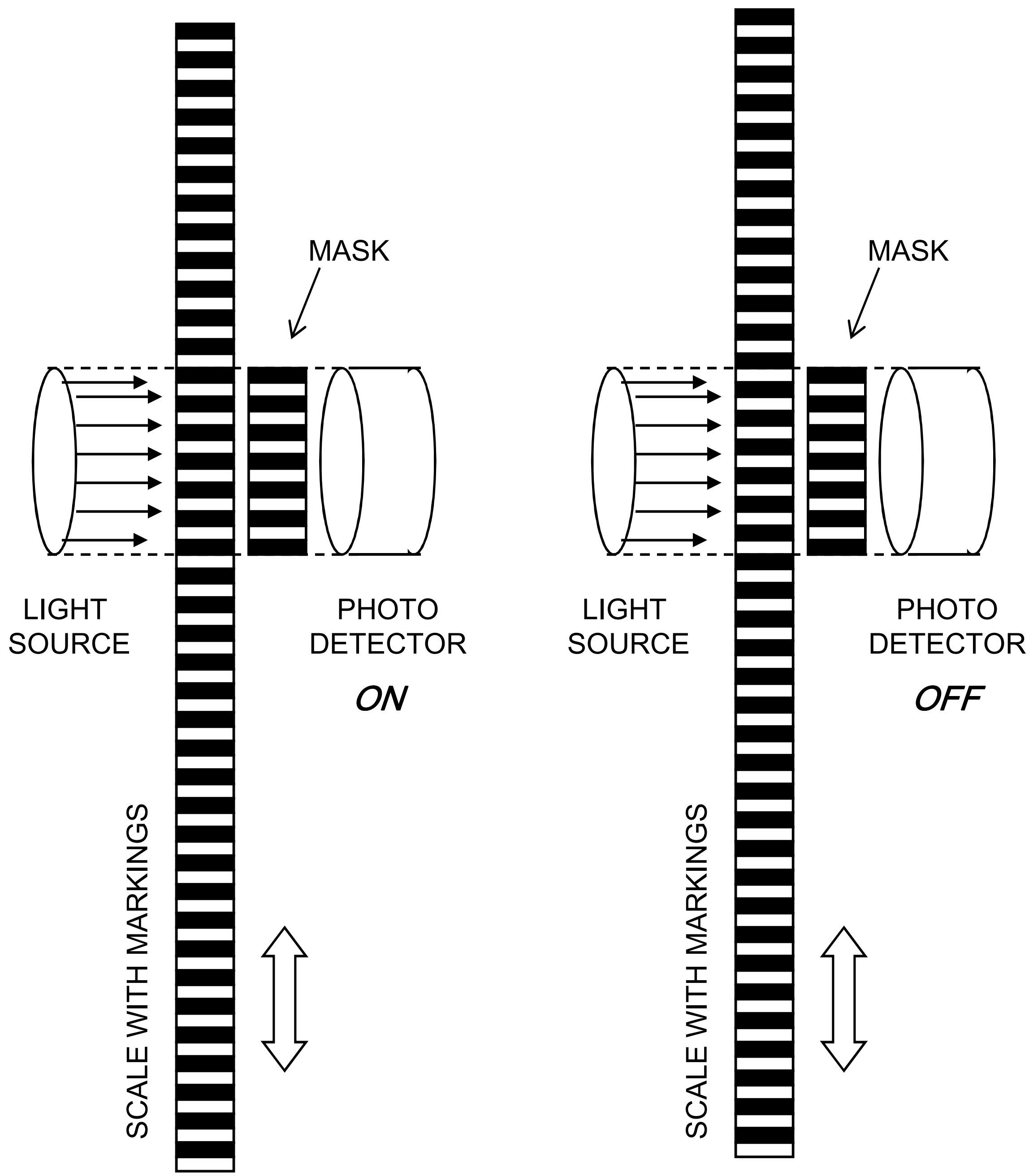
Accuracy is the difference between the actual location and the recovered location.

Repeatability is the variation in the recovered location.

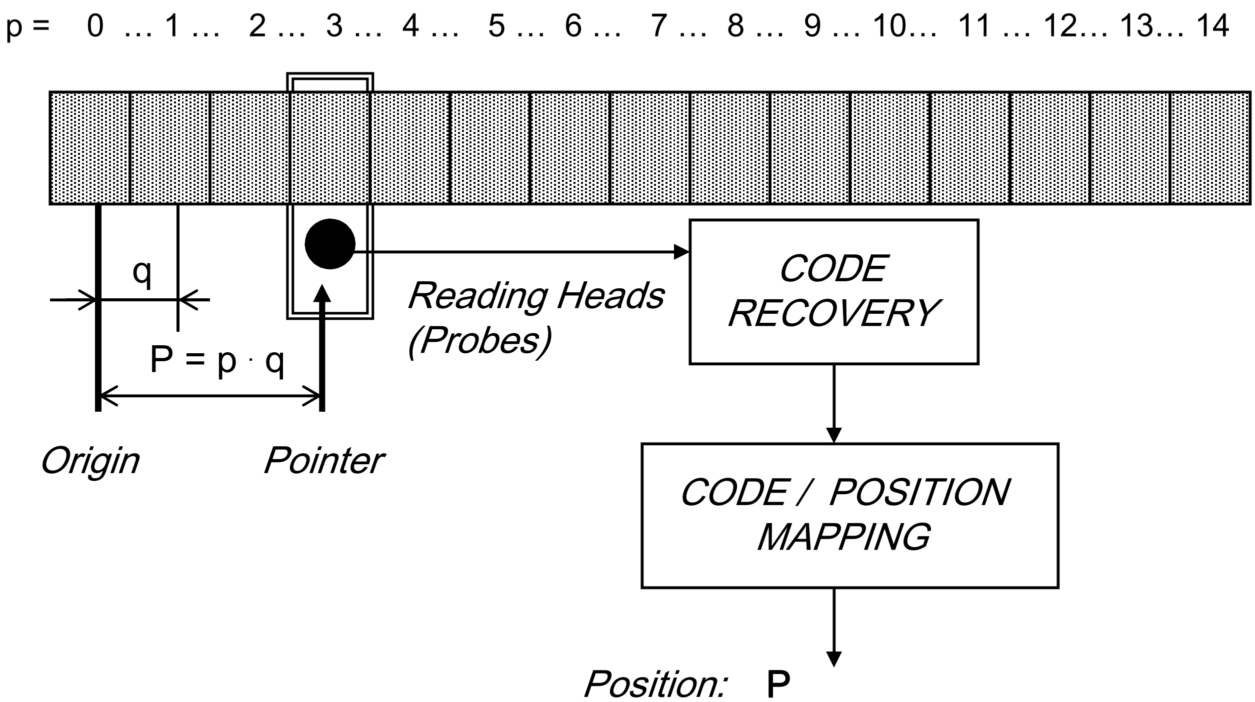
Resolution is the minimum distance that the measurement system can detect.

The repeatability and resolution of most system is better than their accuracy.

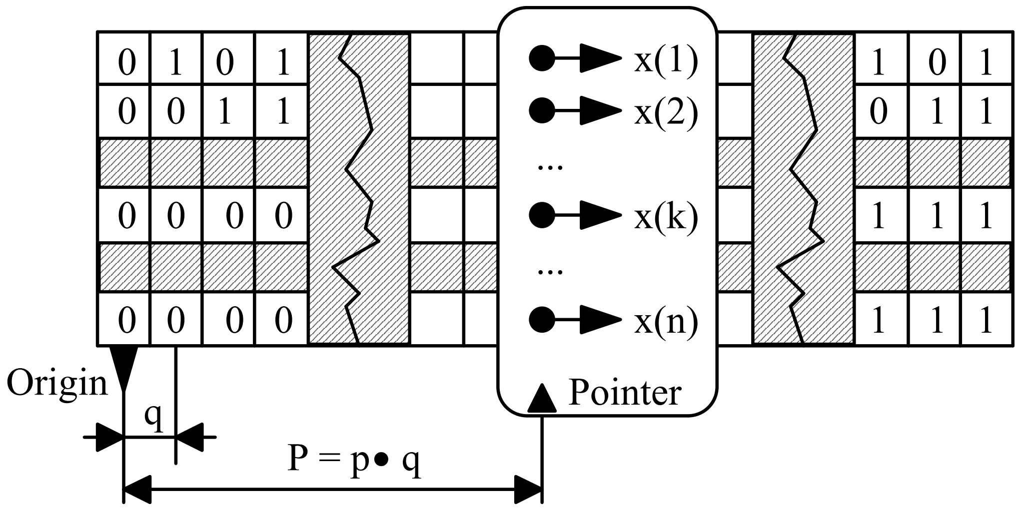
INCREMENTAL OPTICAL ENCODERS



ABSOLUTE POSITION ENCODERS



The quantization intervals on the encoded track are conveniently marked in such a way that the code read for each quantization interval is unique being able to unequivocally identify each Interval. An a priori defined "Code/Position" mapping (mathematical function, or look-up table) is then used to recover the actual position of the quantization interval currently probed .
The most popular codes and associate probe (reading head) technology are binary. However there are now new computer vision, radio beacon, etc. probing technologies that allow multi-valued (polyvalent) marking of the encoded track.

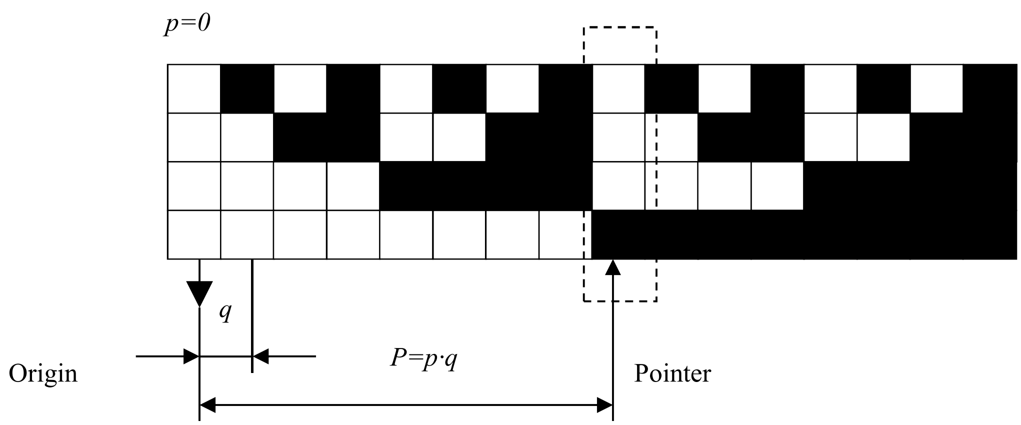


The straightforward approach for the absolute position encoding requires that each quantization interval of a scale be marked with a distinct n-bit code.

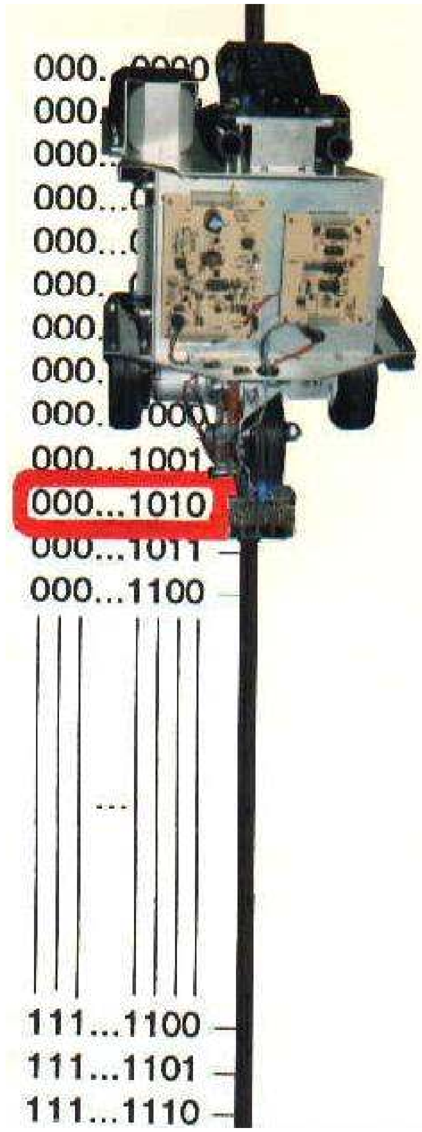
Absolute Natural Binary Encoder.

The position P of the pointer with respect to the origin of this scale is estimated by reading the specific code $\{x(k) \in \{0,1\} \;/\; k=1,\ldots,4\}$ written on the quantization interval currently facing the pointer:

$$P = p \cdot q = \left(\sum_{k=1}^{4} x(k) \cdot 2^{k-1}\right) \cdot q \quad / \quad x(k) \in \{0,1\}$$



Absolute position encoding using a natural 4-bit code



Why natural binary coding cannot be used in practice for absolute position recovery ?

A n-bit code would be needed for each quantization step, resulting in *n* binary tracks in parallel with the guide-path. For instance, the encoding of a 160 m long guide-path with a 0.01 m resolution would need 14 tracks running in parallel with the guide path

Pseudo-Random Encoding

A practical solution allowing absolute position recovery with any desired n-bit resolution while employing only one binary track, regardless of the value of n.

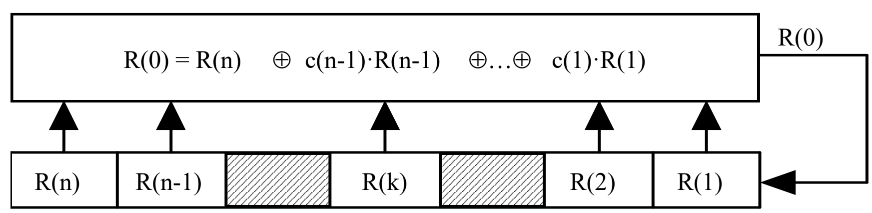
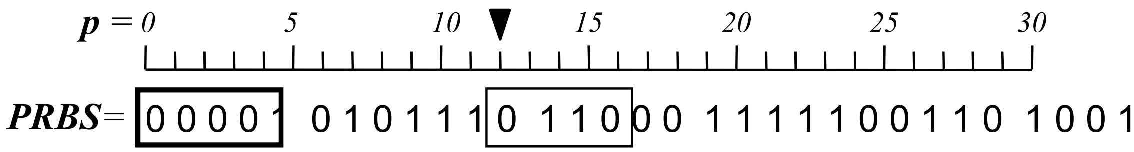


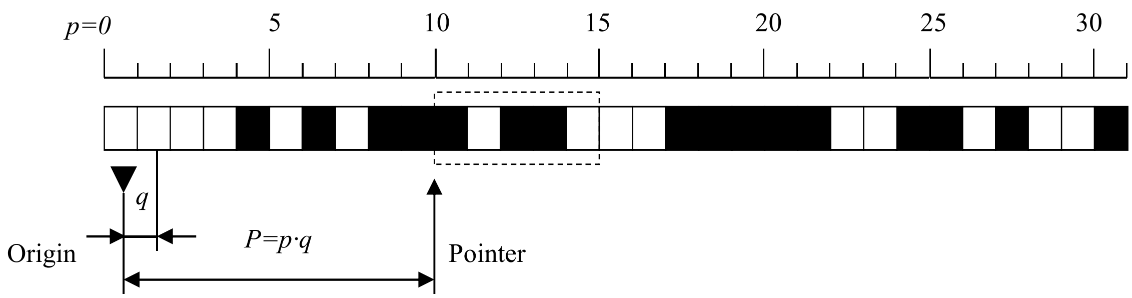
Table 1 Feedback equations for PRBS generation

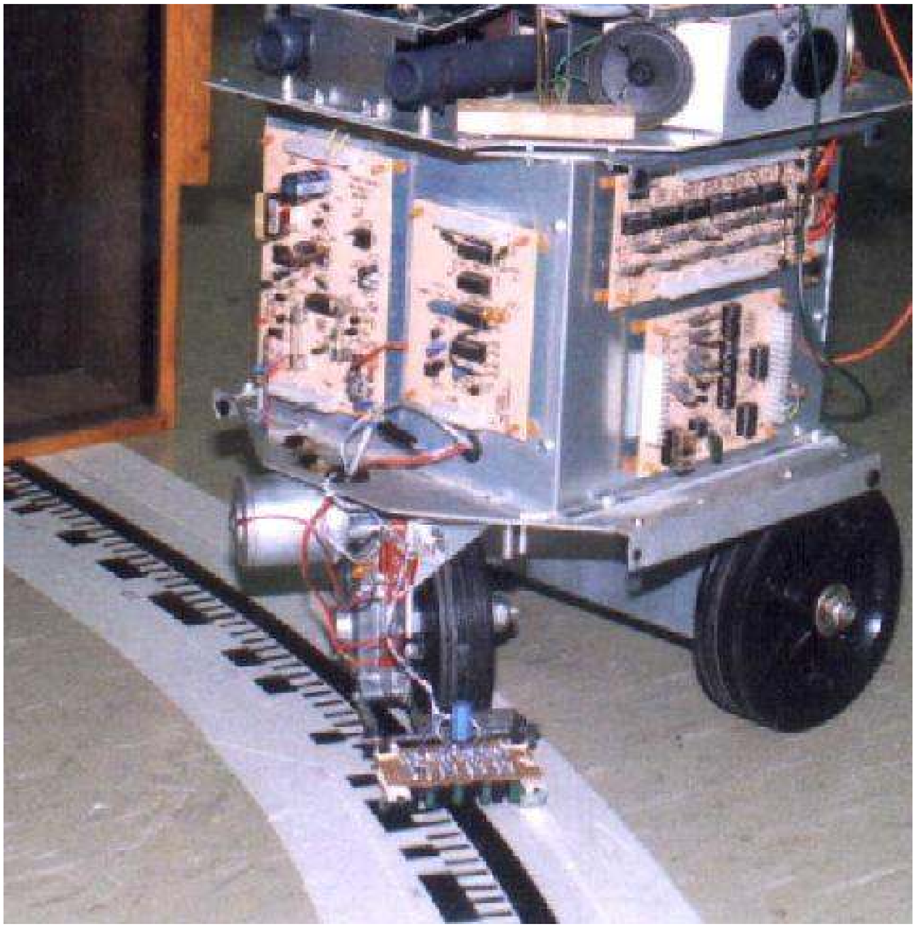
Shift register length *n*	Feedback for direct PRBS R(0)= R(n) ⊕ c(n-1)·R(n-1) ⊕…⊕ c(1)·R(1)	Feedback for reverse PRBS R(n+1)= R(1) ⊕ b(2)·R(2) ⊕…⊕ b(n)·R(n)
4	R(0) = R(4) ⊕ R(1)	R(5) = R(1) ⊕ R(2)
5	R(0) = R(5) ⊕ R(2)	R(6) = R(1) ⊕ R(3)
6	R(0) = R(6) ⊕ R(1)	R(7) = R(1) ⊕ R(2)
7	R(0) = R(7) ⊕ R(3)	R(8) = R(1) ⊕ R(4)
8	R(0) = R(8) ⊕ R(4) ⊕ R(3) ⊕ R(2)	R(9) = R(1) ⊕ R(3) ⊕ R(4) ⊕ R(5)
9	R(0) = R(9) ⊕ R(4)	R(10) = R(1) ⊕ R(5)
10	R(0) = R(10) ⊕ R(3)	R(11) = R(1) ⊕ R(4)



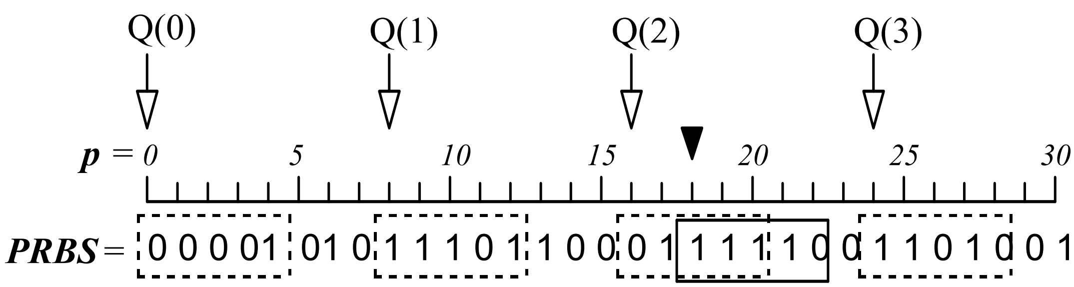
A (2^n -1) term Pseudo-Random Binary Sequences (PRBS) generated by a *n*-bit modulo-2 feedback shift register is used as an *one-bit / quantization-step* absolute code. The absolute position identification is based on the PRBS window property. According to this any *n*-tuple seen through a *n*-bit window sliding over PRBS is unique and henceforth it fully identifies each position of the window.

The figure shows, as an example, a 31-bit term PRBS: 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0,1, 0, 0, 1, generated by a 5-bit shift register. The 5-bit n-tuples seen through a window sliding over this PRBs are unique and represent a 1-bit wide absolute position code.

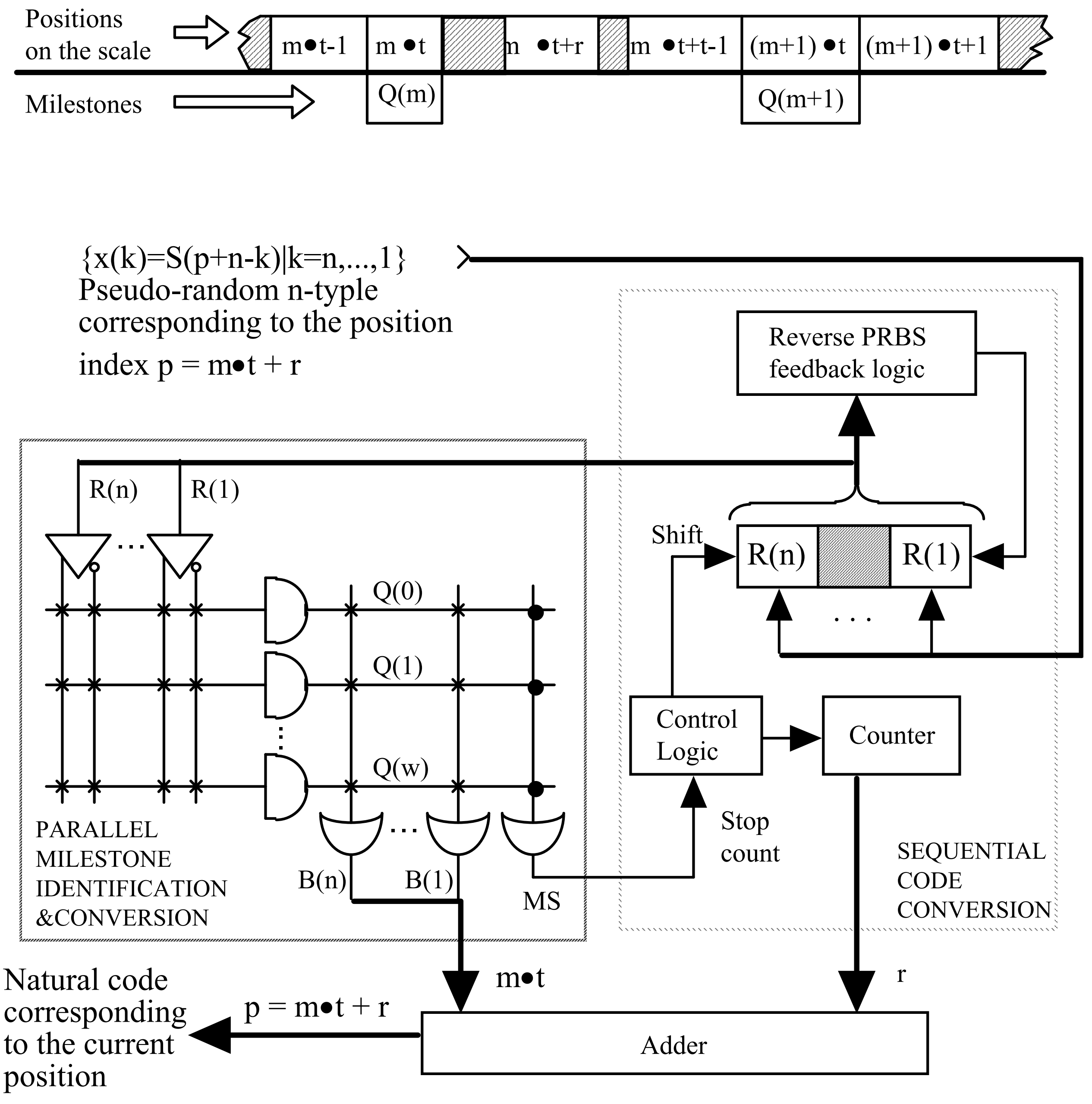




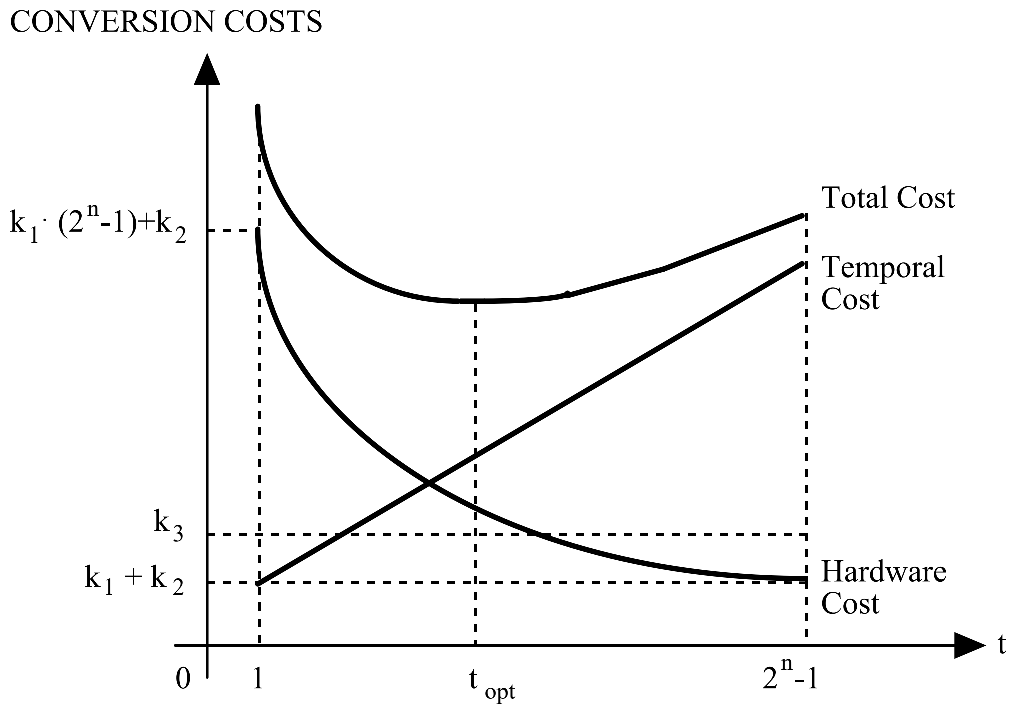
Pseudo-Random Binary Sequence (PRBS) encoded track with one bit per quantization step allows recovery of the absolute position of an optically guided Automated Guided Vehicle (AGV)



Serial-parallel code conversion of the absolute position p=18 on a 31-position PRBS encoded track with four milestones.



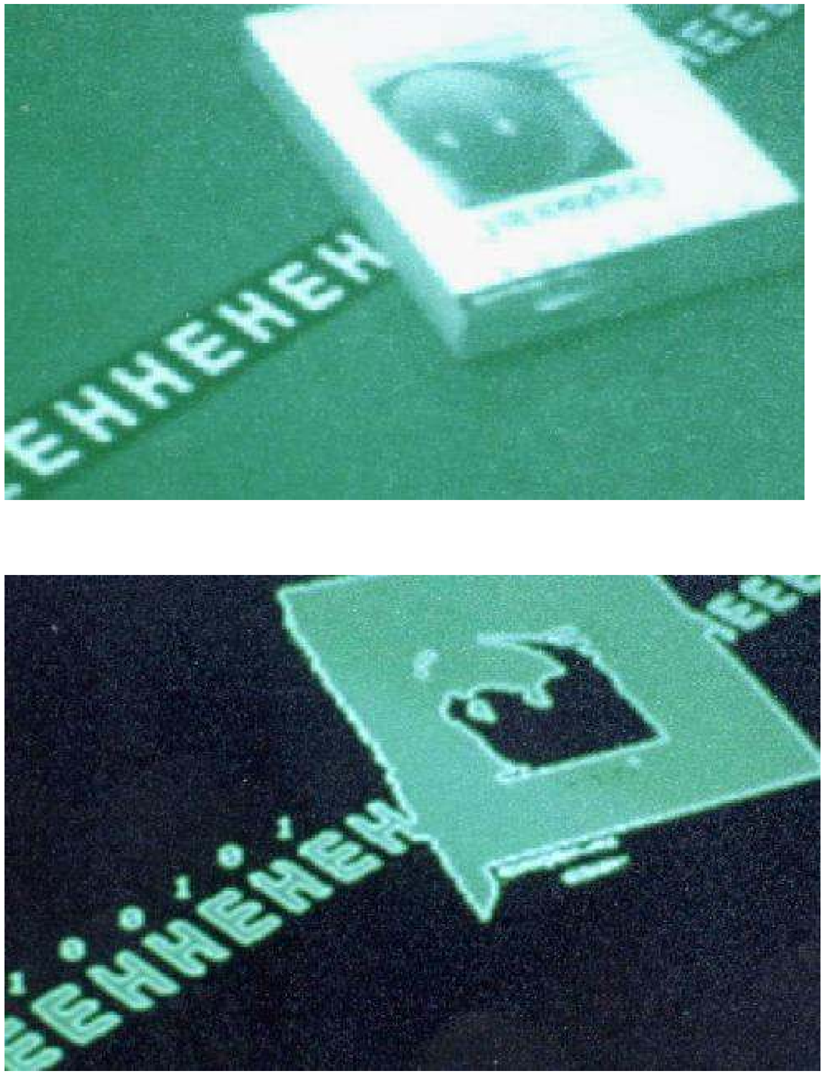
Serial-parallel pseudo-random / natural code conversion algorithm



Serial-parallel code conversion costs as a function of the distance t between milestones. k1 is the equipment cost associated with each milestone, k2 is the basal hardware cost for the serial back-shift operations, k3 is the basal temporal cost for a fully parallel solution, and k4 is the temporal cost associated with each back-shift operation.

Optically guided AGV tracking a PRBS encoded guide path

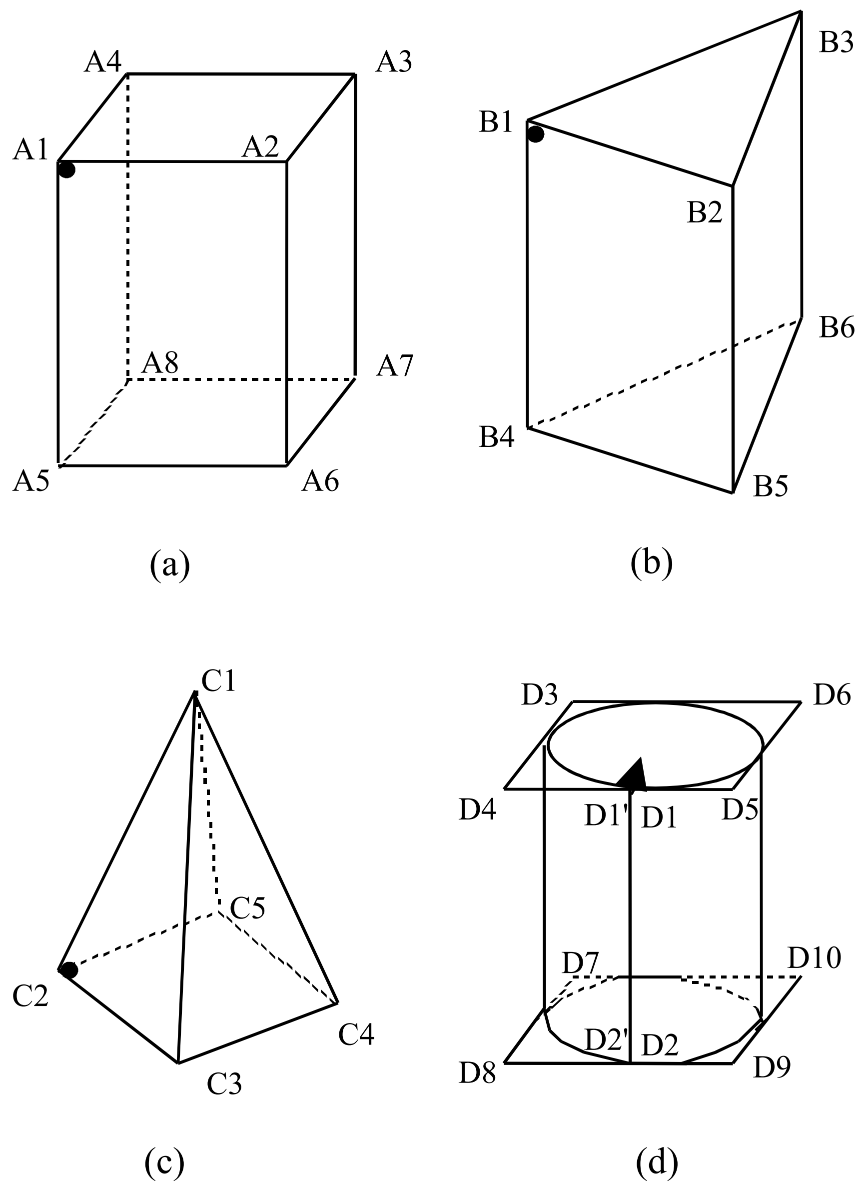
PRBS encoded guide path allows recovery of the absolute position of an AGV using computer vision



Computer vision recognition of the pseudo-random binary code

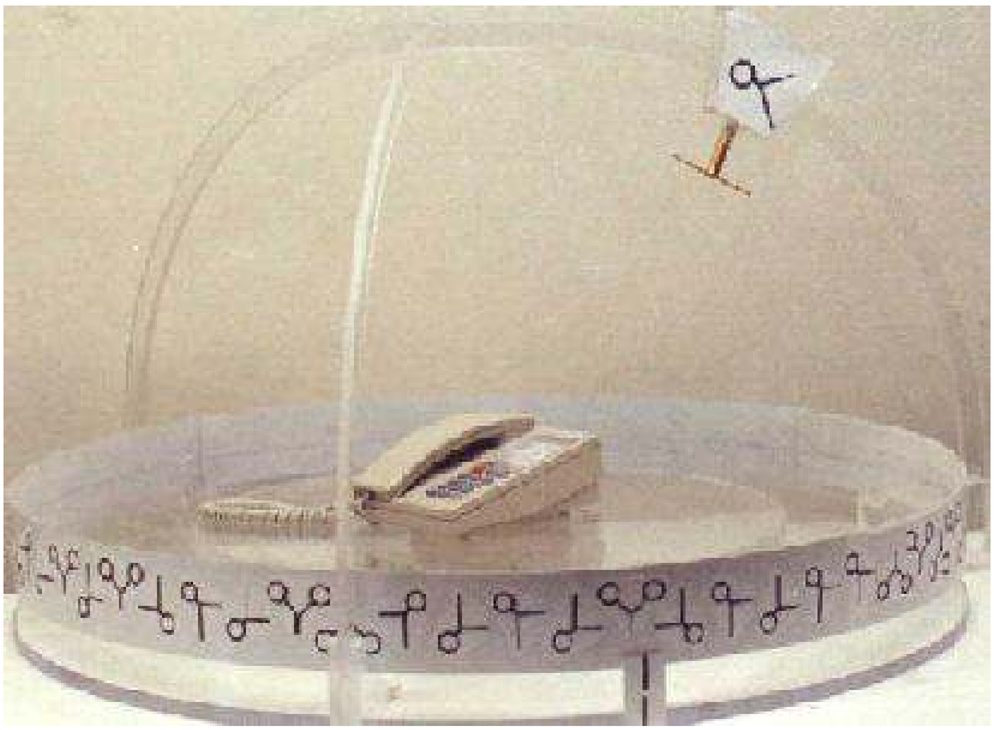
Wall-mounted PRBS encoded guide path allows recovery of the absolute position of the AGV using computer vision

Model Based Object Recognition

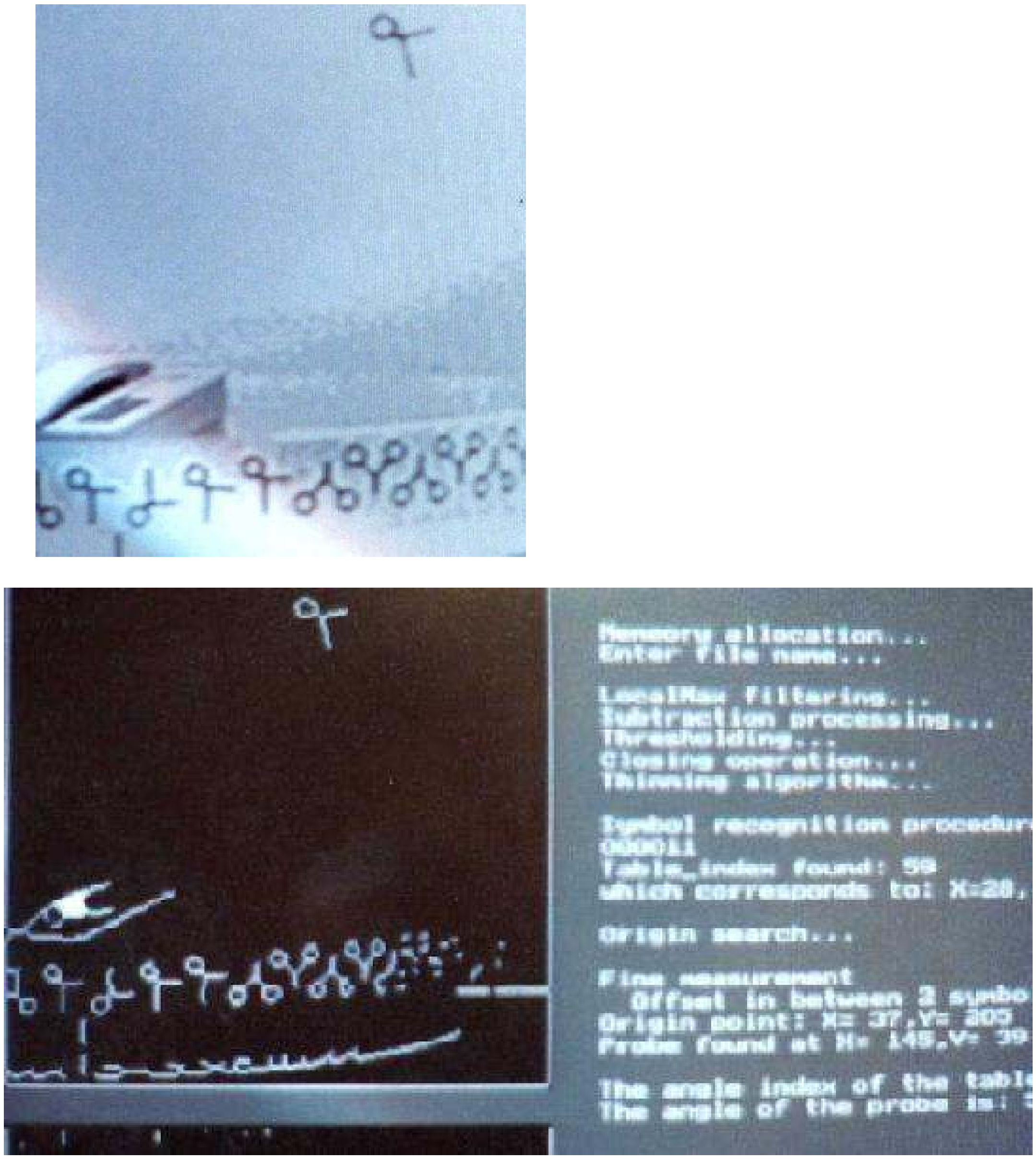


The geometric model definition of four types of 3-D objects:
(a) rectangular parallelepiped, (b) triangular prism,
(c) square pyramid, and (d) right circular cylinder.

Visual Model Based Object Recognition

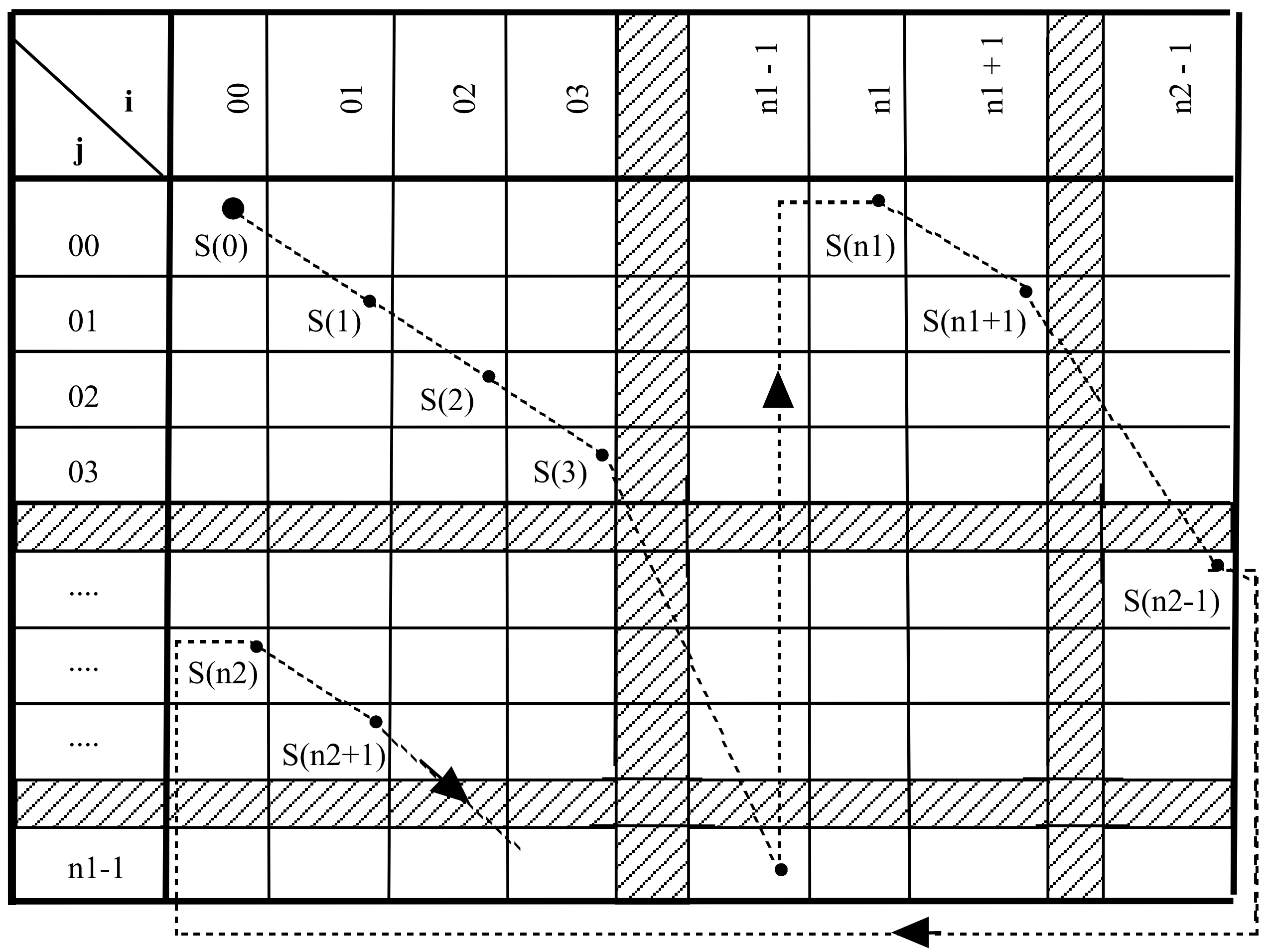


PRBS encoding for computer vision recovery of the 3D position of a probe mapping the electromagnetic–field radiated by a telephone set

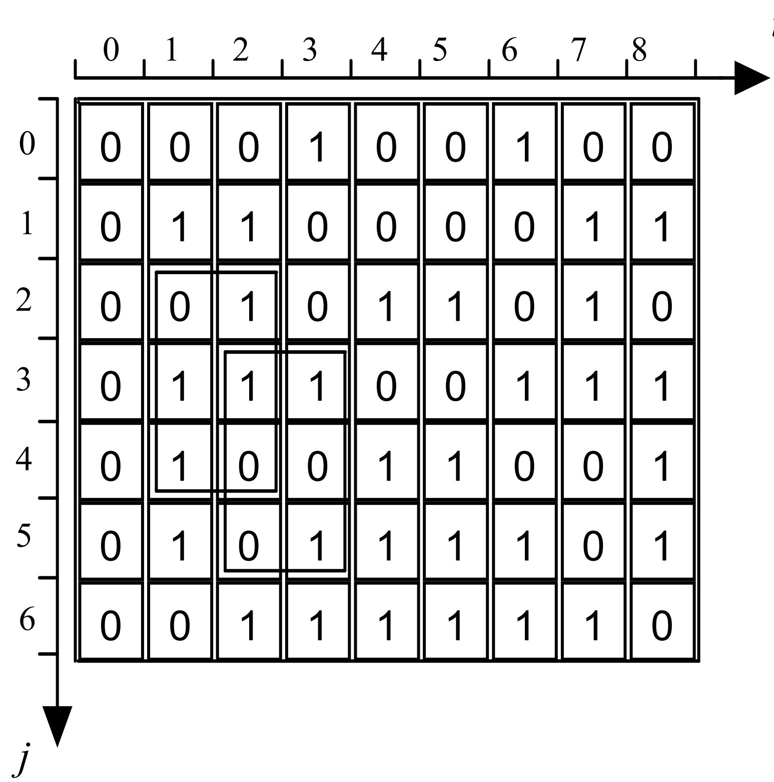


Computer vision recovery of the pseudo-random code

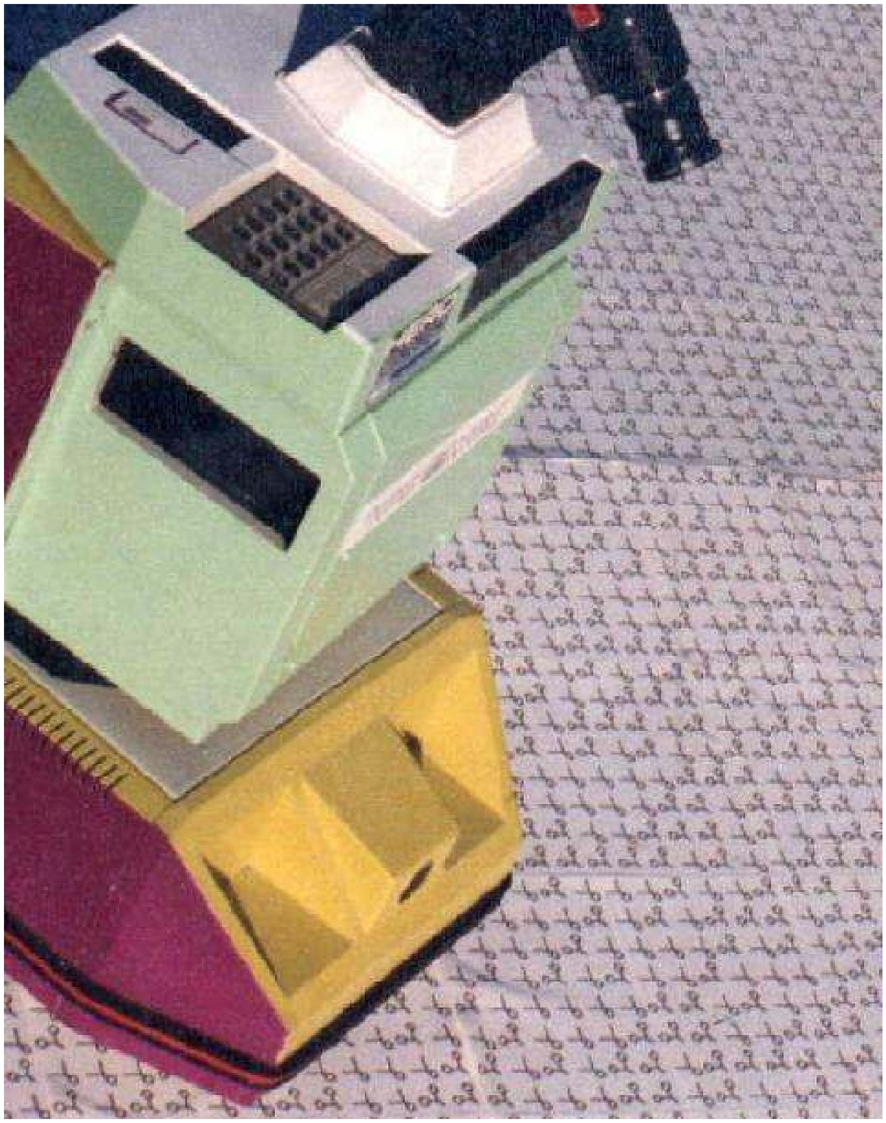
Model-based recognition of a pseudo-random encoded object



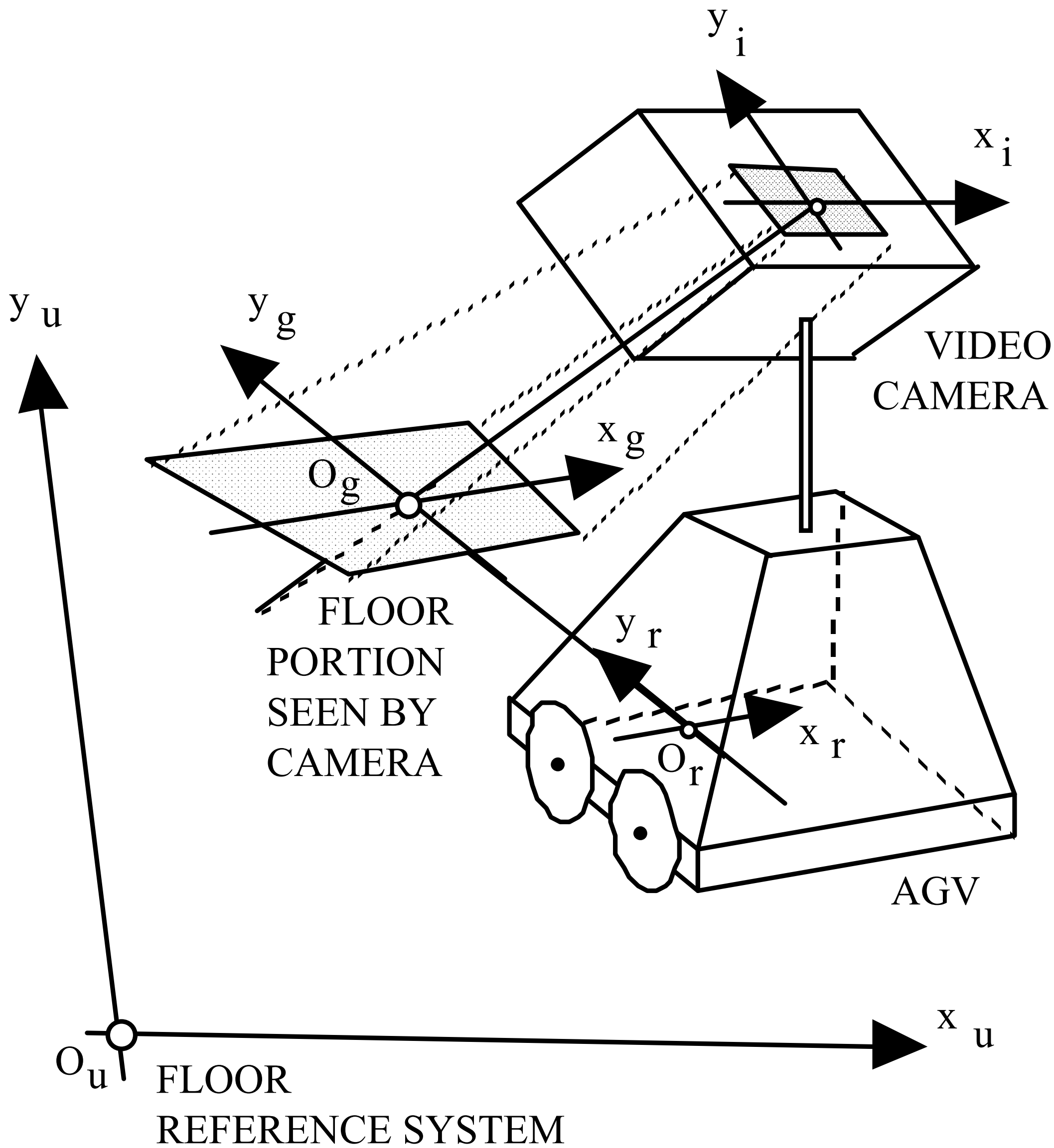
Folding a Pseudo-Random Binary Sequence (PRBS) to produce a Pseudo-Random Binary Array (PRBA)



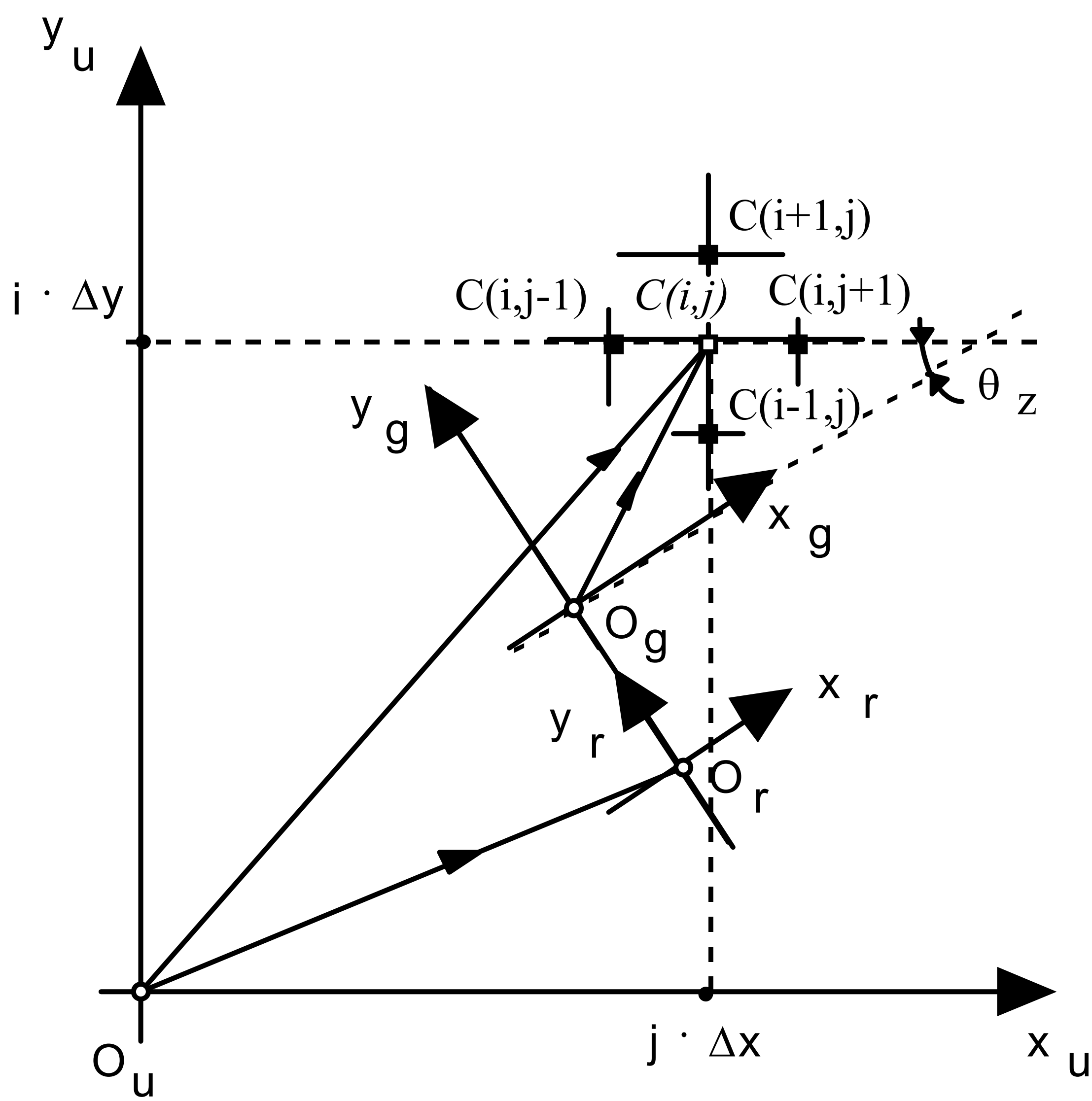
Illustrating the window property in a Pseudo-Random Binary Array (PRBA). The 3-by-2 code seen trough a window on a 7-by-9 PRBA is unique and used as absolute code for the window position (i,j).



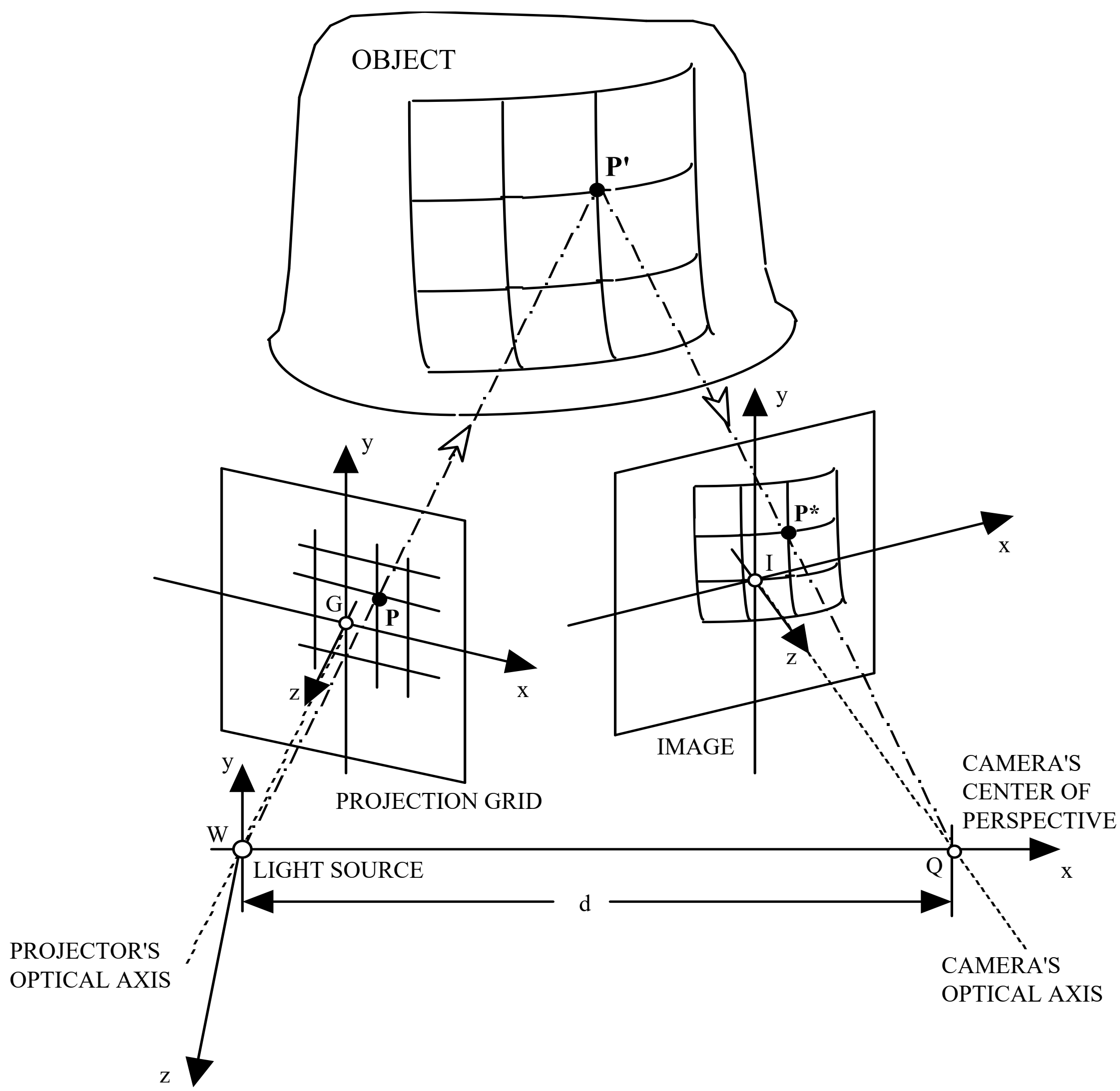
Pseudo-Random Binary Array (PRBA) encoding for the recovery of the 2D absolute position of a free ranging mobile robot using computer vision



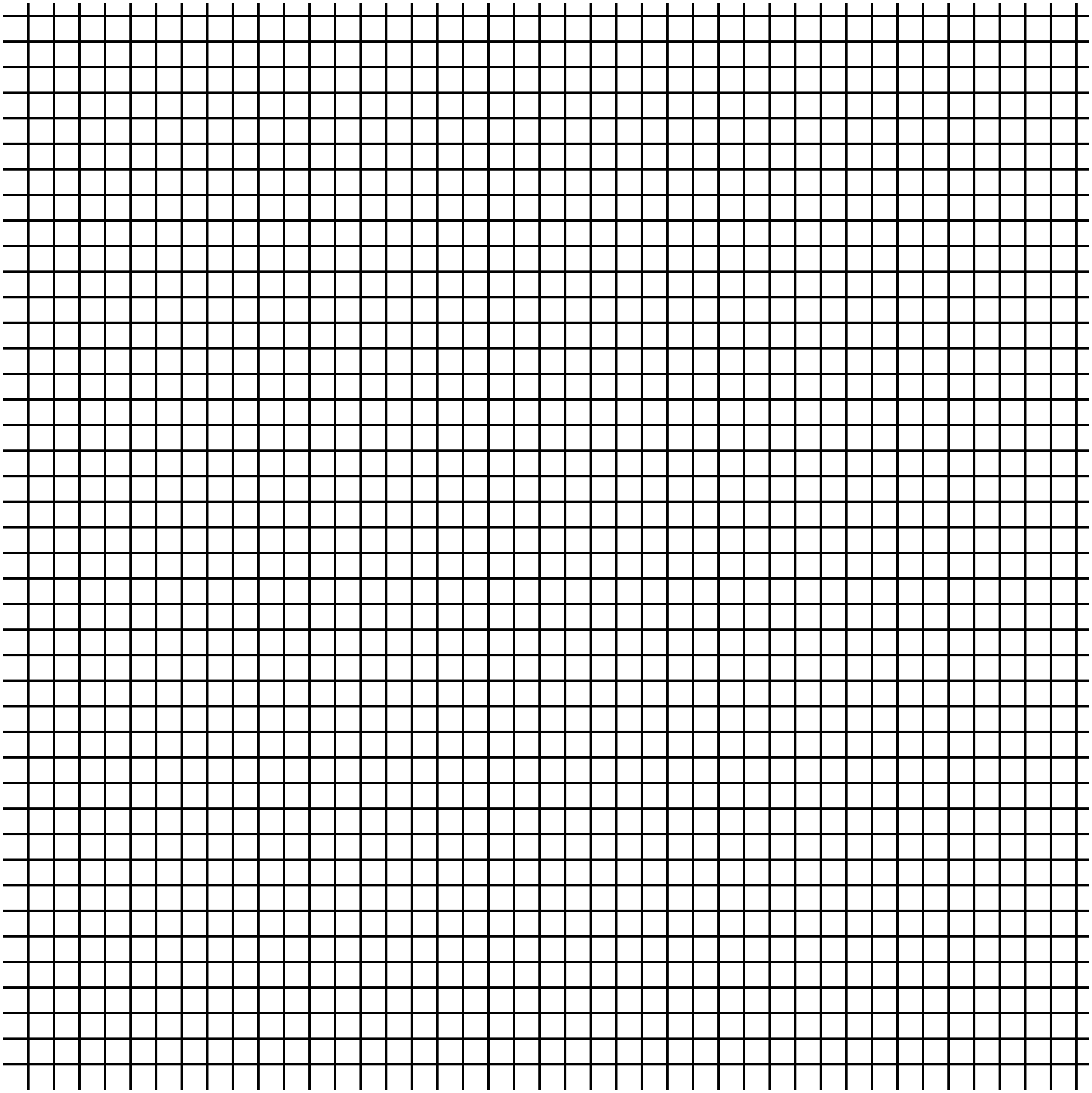
y i
x i
y u
y g
VIDEO
CAMERA
x g
O g
FLOOR
PORTION
SEEN BY
CAMERA
y r
x r
O r
AGV
x u
O u
FLOOR
REFERENCE SYSTEM



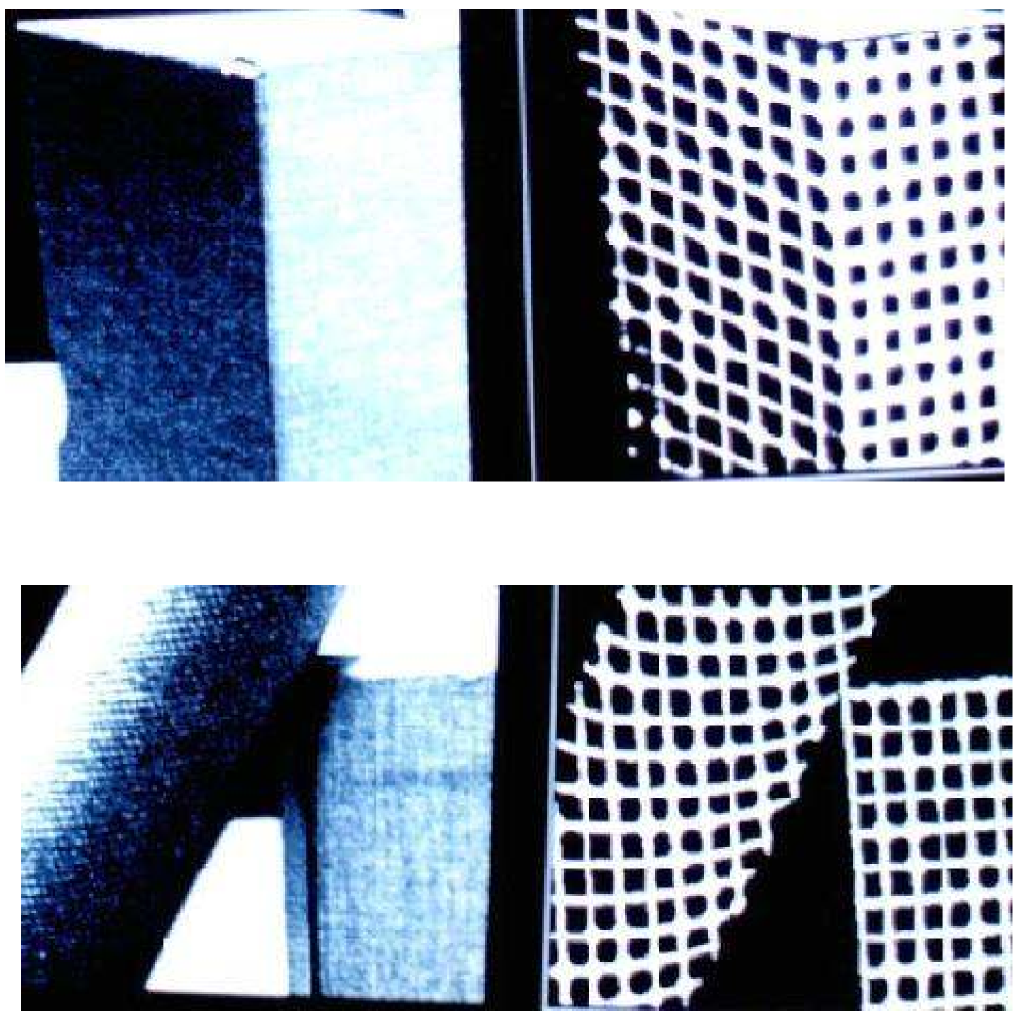
y u
C(i+1,j)
i · Δy
C(i,j-1)
C(i,j)
C(i,j+1)
C(i-1,j)
θ z
y g
x g
O g
x r
y r
O r
O u
j · Δx
x u



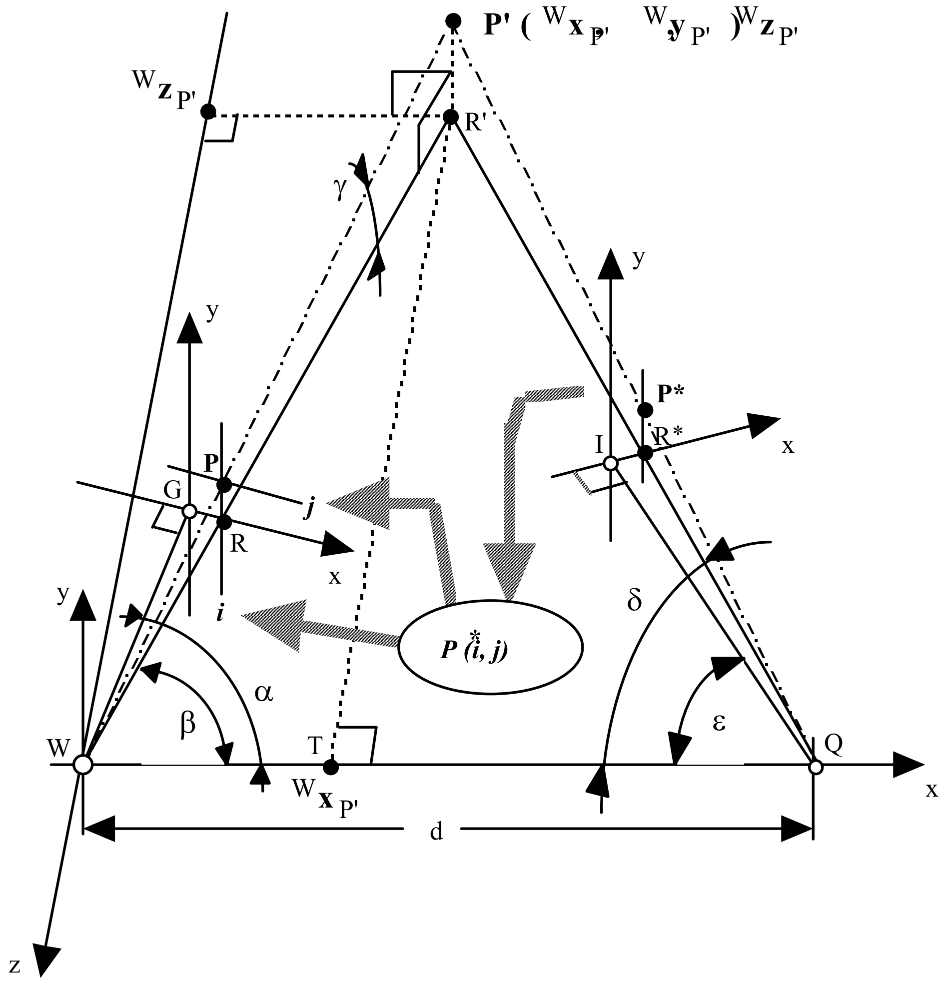
Point identification in pseudo-random encoded structured light



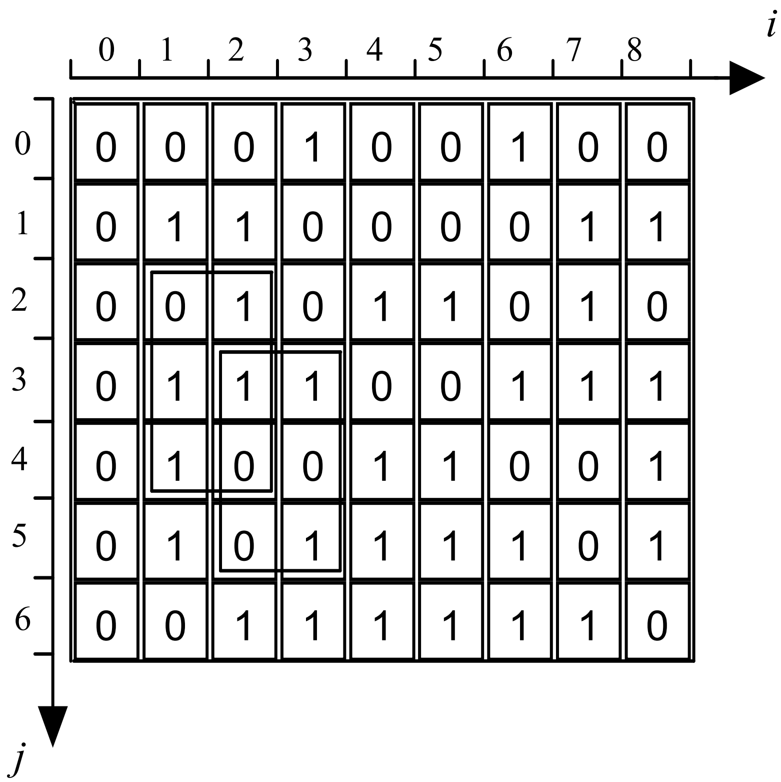
3D Object Recognition Using Structured Light



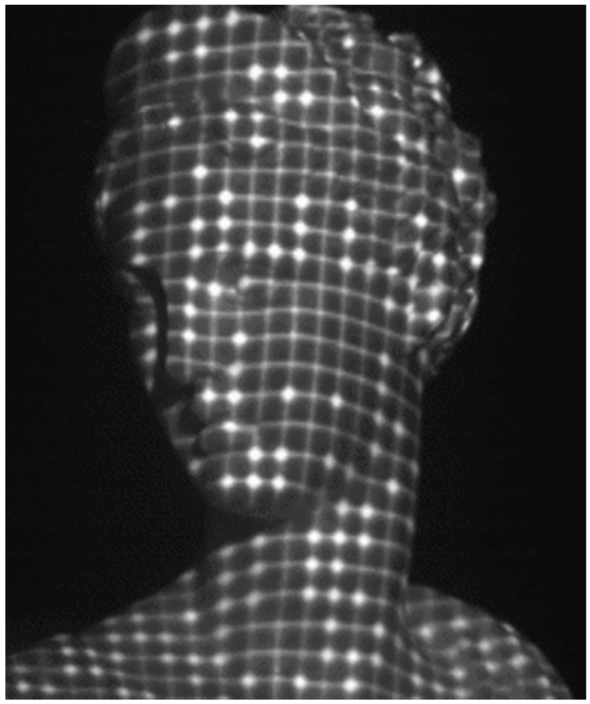
Recovery of the 3D shape of objects using structured light



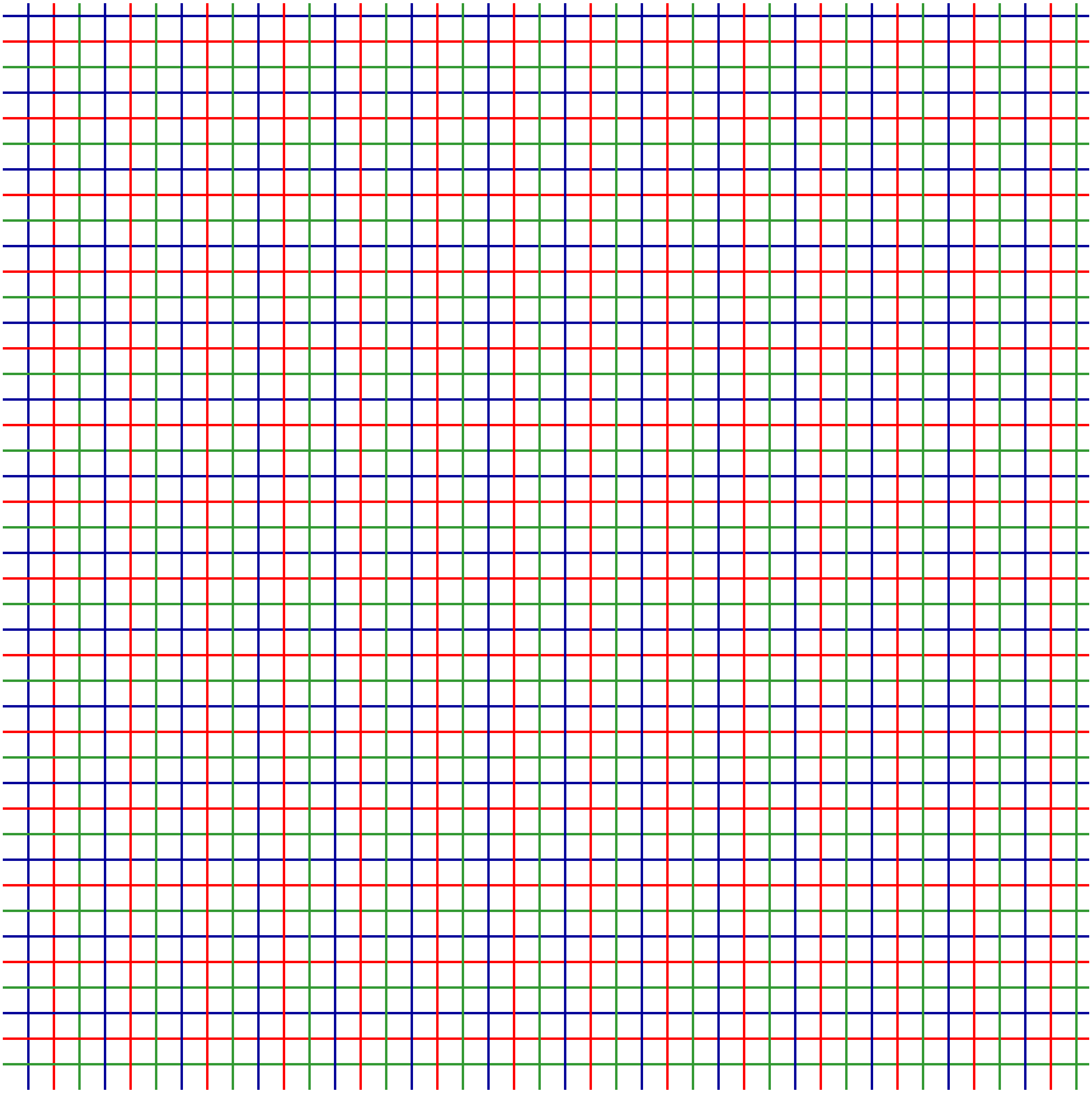
P' (Wx P', Wy P', Wz P')
Wz P'
R'
γ
y
y
x
P*
R*
I
P
G
j
R
x
y
i
P (i, j)
*
α
δ
β
ε
W
T
Q
x
Wx P'
d
z



Illustrating the window property in a Pseudo-Random Binary Array (PRBA). The 3-by-2 code seen trough a window on a 7-by-9 PRBA is unique and used as absolute code for the window position (i,j).



Pseudo-Random Binary Array encoded structured-light grid projected on a 3D object



Pseudo-Random Multi-Valued Sequences (PRMVS)

A "pseudo-random multi-valued sequence" (PRMVS) has multi-valued entries taken from an alphabet of q symbols, where q is a prime or a power of a prime. Such a (q^n -1) -term sequence is generated by an n-position shift register with a feedback path specified by a primitive polynomial

$$h(x) = x^n+h_{n-1}.x^{n-1}+...+h_1.x+ h_0$$

of degree n with coefficients from the Galois field GF(q).

When q is prime, the integers modulo-q form the Galois field GF(q)= {0,1,2,...,p-1} in which the addition, subtraction, multiplication and division are carried out modulo-q.
When q is a power of a prime, q=pm, the integers modulo-q do not form a field and the Galois field elements are expressed as the first q-1 powers of some primitive element, labeled here for convenience by the letter A: GF(q)= $\{0,1,A,A^2,...,A^{q-2}\}$.
The primitive polynomials used for different PRMVS generation depend on the nature of the addition/subtraction and multiplication /division tables adopted for each particular Galois field.

A number of primitive polynomials over GF(q) are given in the next Table for GF(3), GF(4), GF(8), and GF(9). It is obvious that the PRBS is a particular case of PRMVS for GF(2) ={0,1}.

n	q=3	q=4	q=8	q=9
2	x^2+x+2	x^2+x+A	x^2+Ax+A	x^2+x+A
3	x^3+2x+1	$x^3+^{x2+}x+A$	x^3+x+A	x^3+x+A
4	x^4+x+2	$x^{+}+x^2+Ax+A^2$	x^4+x+A^3	x^4+x+A^5
5	x^5+2x+1	x^5+x+A	$x^5+x^2+x+A^3$	x^5+x^2+A
6	x^6+x+2	x^6+x^2+x+A	x^6+x+A	x^6+x^2+Ax+A
7	$x^7+x^6+x^4+1$	$x^7+x^2+Ax+A^2$	$x^7+x^2+Ax+A^3$	x^7+x+A
8	x^8+x^5+2	x^8+x^3+x+A		
9	$x^9+x^7+x^5+1$	x^9+x^2+x+A		
10	$x^{10}+x^9+x^7+2$	$x^{10}+x^3+A(x^2+x+1)$		

The following relations apply:

for $GF(4)= GF(2^2)$: $A^2+A+1=0$, $A^2=A+1$, and $A^3=1$

for $GF(8)= GF(2^3)$: $A^3+A+1=0$, $A^3=A+1$, $A^4=A^2+A$, $A^5=A^2+A+1$, $A^6=A^2+1$, and $A^{7=}1$

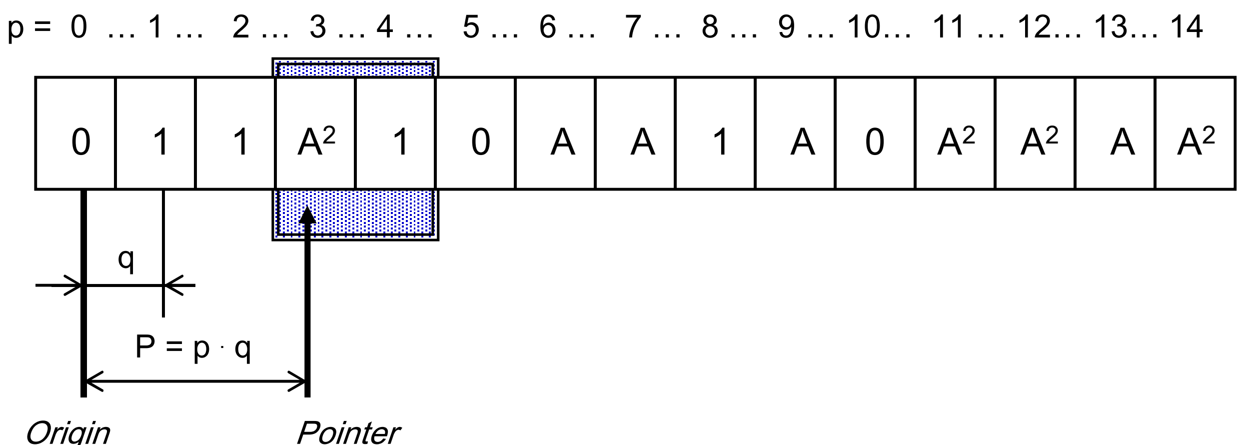
for $GF(9)= GF(3^2)$: $A^2+2A+2=0$, $A^2=A+1$, $A^3=2A+1$, $A^4=2$, $A^5=2A$, $A^6=2A+2$, $A^7=A+2$, and $A^8=1$

According to the PRMVS window property any q-valued contents observed through a n-position window sliding over the PRMVS is unique and fully identifies the current position of the window.

As an example, a two stage shift register, n=2, having the feedback path defined by the primitive polynomial $h(x)= x^2+x+A$ over $GF(4) =\{0,1,A,A^2\}$, with $A^2+A+1=0$ and A3=1, generates the 15-term PRMVS: **{0, 1, 1, A^2, 1, 0, A, A, 1, A, 0, A^2, A^2, A, A^2}.**
Any 2-tuple seen through a 2-position window sliding over this sequence is unique.

A more compact absolute position encoding can be obtained by using *Pseudo-Random Multi-Valued Sequences* (PRMVS) where sequence elements are entries taken from an alphabet with more than two symbols.

Compared to the traditional approach the resulting number of code tracks on the scale at the same resolution decreases proportionally with the size of the alphabet used.



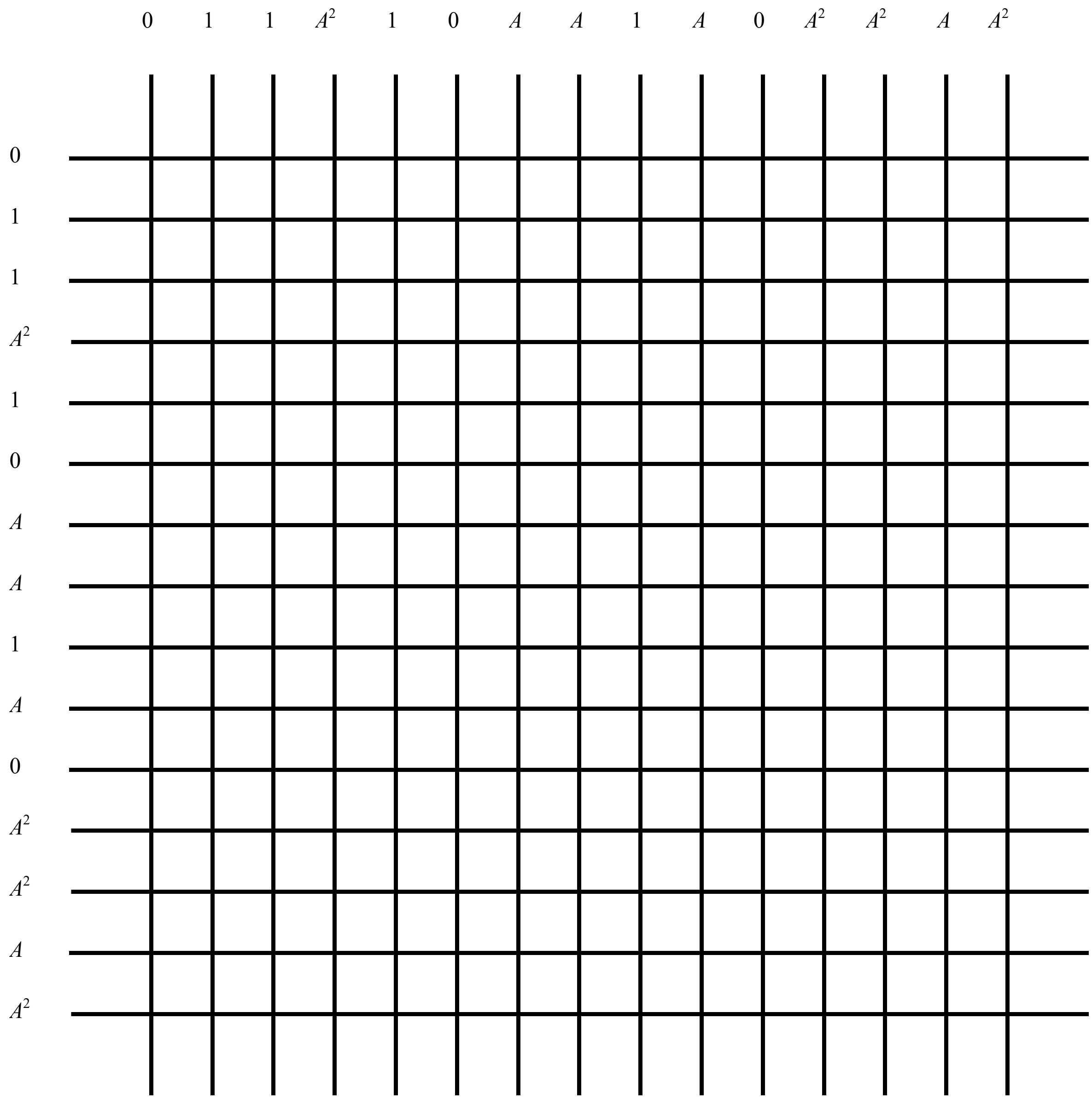
As an example, a two stage shift register, n=2, having the feedback defined by the primitive polynomial h(x)= x^2+x+A over GF(4) ={0,1,A,A^2}, with A^2+A+1=0 and A3=1, generates the 15-term PRMVS ***{0, 1, 1, A^2, 1, 0, A, A, 1, A, 0, A^2, A^2, A, A^2}****. Any 2-tuple seen through a 2-position window sliding over this sequence is unique.*

PRMVS Encoded Grid

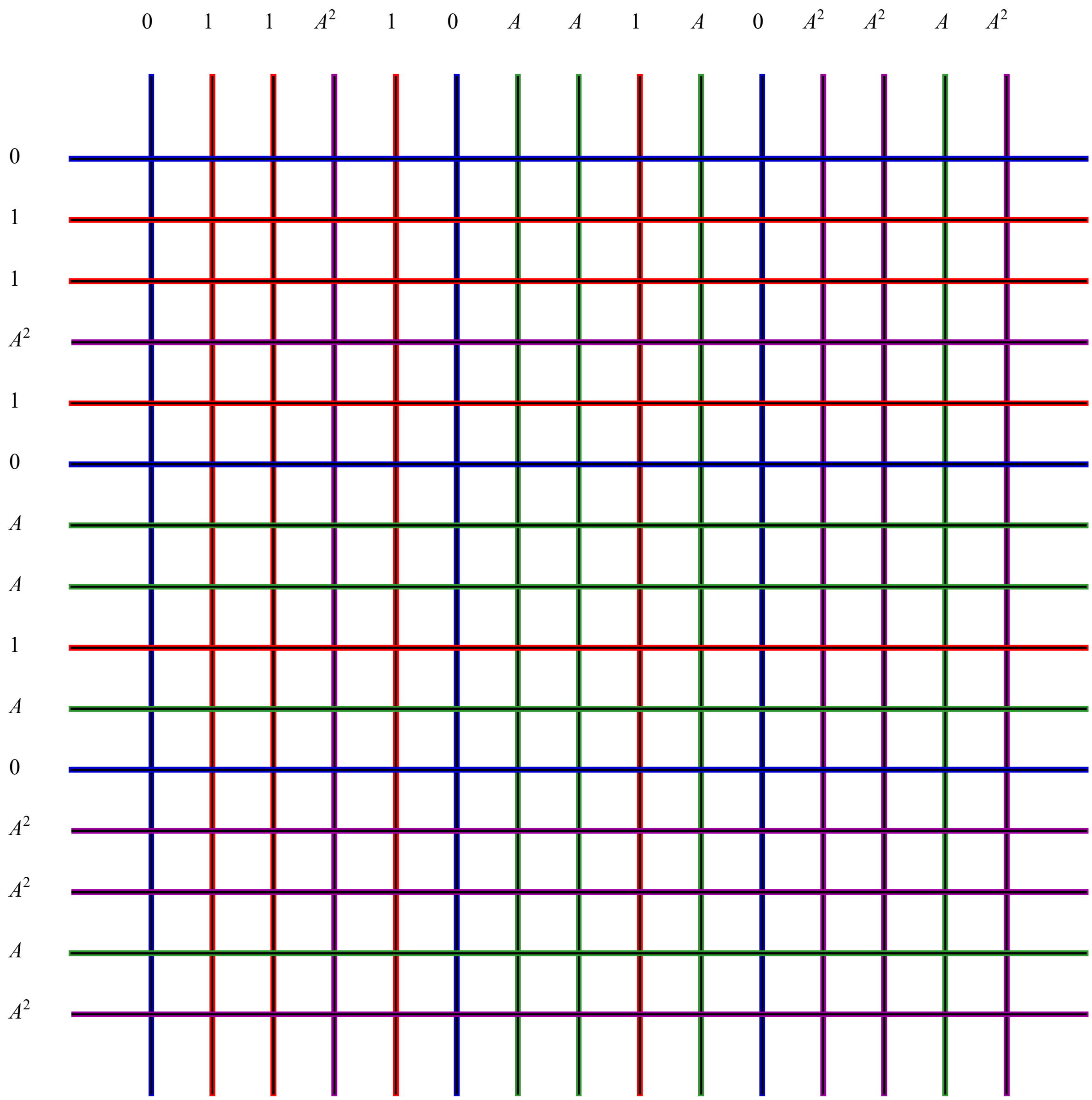
The **rows** are encoded with the terms of a PRMVS $\{X(i)|\ i= 0,1,..., qx^{nx} -1\}$ generated by a **nx** -stage shift register having entries taken from an alphabet of qx symbols.

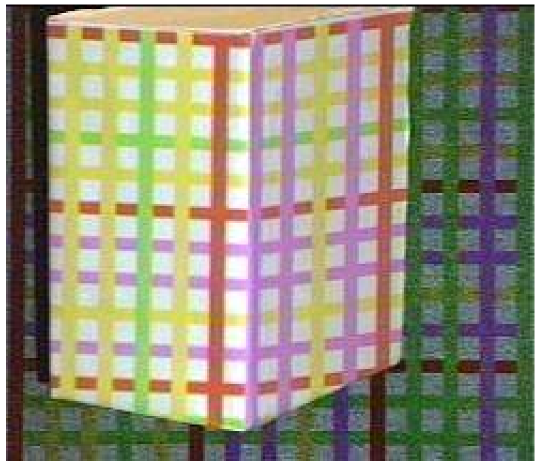
The **columns** are encoded with the terms of a PRMVS $\{Y(j)|\ j= 0,1,..., qy^{ny} -1\}$ generated by a **ny**-stage shift register and having entries taken from an alphabet of qy symbols.

Absolute position recovery of any grid-node of coordinated (i,j) needs to identify a **nx-by-ny window**. The row-index **i** can be recovered if it is possible to identify a nx-tuple containing X(i). The column-index **j** can be recovered if it is possible to identify a ny-tuple containing Y(j).

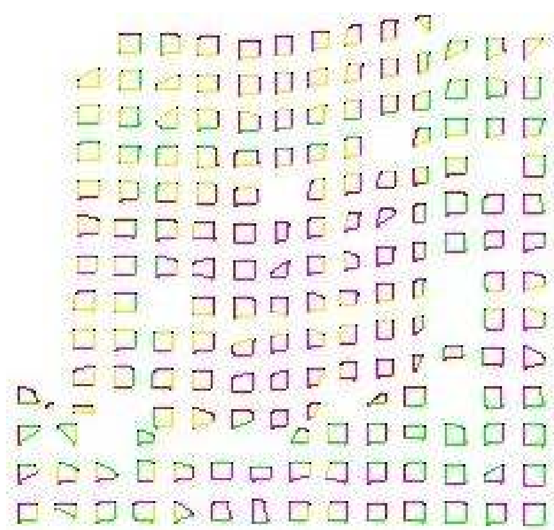


PRMVS grid having 15 row-lines and 15 column-lines encoded with the terms of two PRMVS $\{X(i)=Y(i) \mid i= 0,1,\ldots, q^n -1\}$ where $q=4$ and $n=2$, defined over $GF(4)=\{0,1,A, A^2\}$. Absolute position recovery needs to identify a 2-by-2 window in this case. The row-index i of a given grid node (i,j) can be recovered if it is possible to identify the X(i) and X(i+1) [or X(i-1)] associated with two adjacent row lines. The column-index j of a given grid node (i,j) can be recovered if it is possible to identify the Y(j) and Y(j+1) [or Y(j-1)] associated with two adjacent column lines.


0 1 1 A^2 1 0 A A 1 A 0 A^2 A^2 A A^2
0 1 1 A^2 1 0 A A 1 A 0 A^2 A^2 A A^2



Pseudo-Random Multi Valued Sequence (PRMVS) opportunistically color encoded structured light grid projected on a cube



Recovered corner points at the intersection of grid line edges

Pseudo-Random Multi Valued Sequence (PRMVS)
structured-light grid projected on a 3D object

WHERE AM I ?

... GLAD TO BE HERE .

THANK YOU !