UDC: 512.624:514.144
Conference paper

FINITE PROJECTIVE PLANES AND HAMMING CODES

Flamure Sadiki[1*], Alit Ibraimi[1], Krutan Rasimi[1], Ylldrita Seferi[1]

[1*]*Department of Mathematics, Faculty of Natural Science and Mathematics, University of Tetova, RNM*

**Corresponding author e-mail: flamure.sadiki@unite.edu.mk*

Abstract

This paper is a short survey of projective geometry, history of Hamming codes and the relationship between them and projective planes. The projective plane of order n, over a finite field $(F_p = \{0,1,2,..., p-1\},+,*)$, which has arithmetic done $\bmod p$, denoted by $PG(n,p)$, is the set of all subspaces of vector space F_p^n. It can be endowed with the distance function $f(U,V) = \dim(U) + \dim(V) - 2\dim(U \cap V)$, which turns $PG(n,p)$ into a metric space. A (n,M,d) code in projective space is a subset of $PG(n,p)$ of size M such that the distance between any two code-words is at least d. The first error correction code, the Hamming code, is intrinsic to the projective plane of order 2 over F_2. A connection between planes and codes is given by construction of Hamming code related to $PG(2,2)$ and we generalize them to $PG(3,2)$ using Hamming and Generator matrix. The codes constructed in this way are called projective codes.

Keywords: Projective plane, Finite field, Hamming codes, Hamming Matrix, Generator Matrix

1. Introduction

Let F_p be the finite field of order p, which in fact is $(F_p = \{0,1,2,..., p-1\},+,*)$ with arithmetic done by $\bmod p$ and let W be an arbitrary (fixed) vector space of dimension n over F_p. Since W is isomorphic to F_p^n, in what follows one can assume that W is in fact F_p^n. The projective space of order over n, denoted herein by $PG(n,p)$, is the set of all the subspaces of W, including $\{0\}$ and itself. In fact it is the set of all subspaces of vector space F_p^n, including their isomorphism. Given a nonnegative integer $k \leq n$, the set of all subspaces of W that have dimension k is known as a Grassmannian, and usually denoted by $G_p(n,k)$. Thus

$$PG(n,p) = \bigcup_{0 \leq k \leq n} G_p(n,k)$$

$$|G_p(n,k)| = \begin{bmatrix} n \\ k \end{bmatrix} \overline{\overline{def}} \frac{(p^n - 1)(p^{n-1} - 1) \cdot ... \cdot (p^{n-k+!1} - 1)}{(p^k - 1)(p^{k-1} - 1) \cdot ... \cdot (p - 1)}$$

where $\begin{bmatrix} n \\ k \end{bmatrix}$ is the p -ary Gaussian coefficient. It turns out that the natural measure of distance in $PG(n,p)$ is given by $f(U,V) = \dim(U) + \dim(V) - \dim(U \cap V),\ \forall U,V \in PG(n,p)$. It is well known that the function above is a metric, thus both $PG(n,p)$ and $G_p(n,k)$ can be regarded as metric spaces. Given a metric space, one can define codes. We say that $C \subseteq PG(n,p)$ is a (n,M,d)code in projective space if $|C| = M$ and $f(U,V) \geq d$ for all U,V in C. If a code is contained in for some, we say that is a code. If a (n,M,d)code C is contained in $G_p(n,k)$for some k, we say that C is a (n,M,d,k)code. The(n,M,d), respectively (n,M,d,k), codes in projective space are akin to the familiar codes in the Hamming space, respectively (constant-weight) codes in the Johnson space, where the Hamming distance serves as the metric. There are, however, important differences.

2. Projective plane

For centuries most geometry was done according to his axioms. The Parallel Line Axioms, however was a source of contention for many. The axioms states:

Given a line *l* and a point *p* not on that line, then there is a unique line containing *p* which is parallel to *l*.

It was not until the 1800's that Bernhard Riemann suggested that there may not be any parallel lines at all. It was from this idea that projective plane geometry was developed.

A projective plane is a collection of points and lines which satisfy the following three properties:

- For any two lines, there is a unique point of intersection.
- For any two points, there is a unique line containing them.
- There are 4 points of which no three are colinear.

One geometric way to describe a projective plane is to start with a 3-dimensional Euclidean space, and then to project through the origin. The 1-dimensional subspaces of the original 3-space are the points of the projective plane. The 2-dimensional subspaces of the original 3-space are the lines of the projective plane.

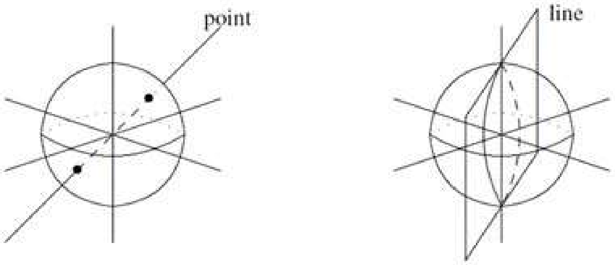


Figure 1. Point and line

3. Fields

Projective planes can be given coordinates. A plane that has coordinates is called a coordinatized plane. For the planes discussed in this paper, the coordinates come from an algebraic structure called a field.

A *Field* is a set F with binary operations +, * s.t.
$a+b=b+a$(addition is commutative).
$(a+b)+c=a+(b+c)$(addition is associative)
$a+0=0+a=a$ (0 is the additive identity)
$a+(-a)=0$ (every element has an additive inverse).
$a*b=b*a$ (multiplication is commutative).
$(a*b)*c=a*(b*c)$(multiplication is associative)
$a*1=1*a=a$ (1 is the multiplicative identity)
$a*(a^{-1})=1$ (every element has a multiplicative inverse).
$a*(b+c)=a*b+a*c$ (multiplication is distributive).
Finite fields exist for prime powers p^k. A finite field (F_p = { 0,1, 2,….p-1},+,*)has arithmetic done *mod p*: Binary arithmetic is done over the field of order 2, which is denoted F_2.

Example: $PG(2,2)$

Notice that $PG(2,2)$ has 7 points and 7 lines. There are no parallel lines, each pair of lines shares a unique point, and each pair of points are contained by a unique line. Each line contains 3 points and each point lies on 3 lines. This plane, sometimes called the Fano Plane, is coordinatized by F_2 (the field of order 2). It is also worth noting that the sum of two points is the third point on that line.

Simple and Coordinatized Fano Plane is given in the Figure 2:

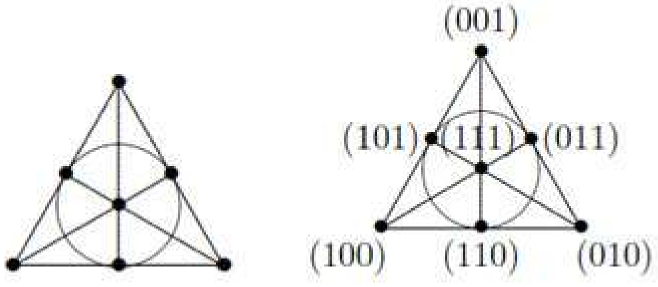


Figure 2. Simple and Coordinatized Fano Plane

4. The Coding Problem

All considerations in coding theory are based on the following communication model. A sender wants to transmit data to a recipient. These data are transmitted via a channel that, despite any amount of care, might not transmit the data unaltered - there might be random noise, usually due to circumstances beyond the sender's control. Probably everybody has

experienced the irritation caused by poor reception due to 'atmospheric noise' during a favorite TV program.
The sender encodes data d into a message *c* (also called a codeword); this codeword will be transmitted over the channel. The recipient decodes the message and tries to detect whether errors have occurred or not. If one uses 'error -correcting codes', then the original data can again be reconstructed.

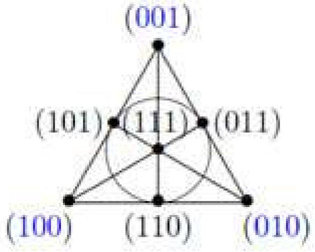
4.1.Hamming code

The message is always a binary string of length *n*, hence an element of the vector space. The problem we want to study can be described as follows. The channel adds to the transmitted vector *c* (the 'message') an error vector *e*, so the recipient receives the vector *x = c + e.* The recipient's aim then is to decode *w*, that is, to determine the error vector to reconstruct *c* from *w*.
The Hamming Code consists of two matrices: The Hamming matrix and the Generator matrix.

$$H=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad G=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

A Connection between plane and code:

$$H=\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}, G=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{bmatrix} \leftrightarrow$$



The Hamming Code and the projective plane of order 2, the $PG(2,2)$ are closely related. The points of the projective plane make up the columns of the Hamming Matrix H .

For convenience, we arrange H in block form. Let I be the 3 x 3 identity matrix. Let P be the 3 x 4 matrix with the remaining points of PG(2, 2) as its columns $H=\begin{bmatrix} P & I \end{bmatrix}$. Therefore G is of the form $G=\begin{bmatrix} I \\ -P \end{bmatrix}$. Note that in G, I is the 4 x 4 identity matrix.

Example

A code from $PG(3,2)$ consist of 13 points and 13 lines over the field of order 3. To construct a code from $PG(3,2)$, let I be the 3 x 3 identity matrix which corresponds to three of the points of the plane. Then let P be the 3 x 10 matrix with the remaining points of $PG(3,2)$ as its columns. Hamming and Generator matrix, in this case, have the forms:

$$H = \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 2 & 1 & 2 & 1 & 2 & 0 & 1 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 & 1 & 1 & 2 & 2 & 0 & 0 & 1 \end{bmatrix} = [P \ I]$$

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 0 & 0 & 2 & 1 & 2 & 1 & 2 & 1 \\ 2 & 1 & 2 & 1 & 0 & 0 & 2 & 2 & 1 & 1 \end{bmatrix} = \begin{bmatrix} I \\ -P \end{bmatrix}$$

related to the coordinated projective plane:

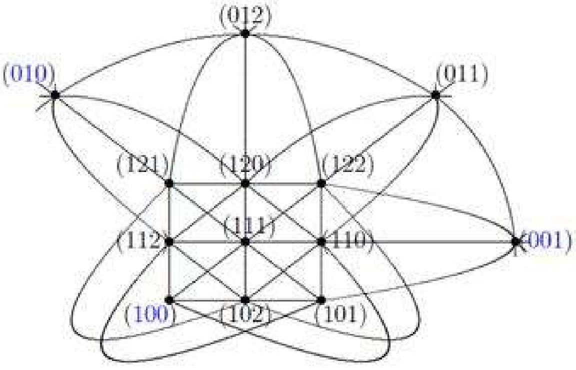


Figure 3. The coordinated projective plane

In G,the matrix I is the 10×10 identity matrix. H and G provide a code with a vocabulary of size 3^{10} consisting of words which are strings of length 10 with an alphabet from F_3. The code words are length 13 and are attained by multiplication by G: $c = Gw$. The code can determine the location and scalar value of a single error.

Using the Hamming Matrix and Generator Matrix we can see how the encoding and error detection work. We encode a word w with multiplication by G: $c = Gw = \begin{bmatrix} I \\ -P \end{bmatrix} w = \begin{bmatrix} w \\ -Pw \end{bmatrix}$, We determine the position and scalar value of a single error with multiplication by *H*.

Case I (no error): $r = c$

$$H\begin{bmatrix} w \\ -Pw \end{bmatrix} = [P \quad I] \cdot \begin{bmatrix} w \\ -Pw \end{bmatrix} = [Pw + (-P)w] = [0]$$

Case II (one error): $r = c + e_i = \begin{bmatrix} w \\ -Pw \end{bmatrix} + s[e_i]$, where $s \in F_3$ and $[e_i]$ is a vector with 1 in the i-th positions and 0 elsewhere.

$$H\left(\begin{bmatrix} w \\ -Pw \end{bmatrix} + s[e_i]\right) = [P \quad I] \cdot \left(\begin{bmatrix} w \\ -Pw \end{bmatrix} + s[e_i]\right) = [Pw + (-P)w] + sH[e_i] = sH[e_i]$$

Since $sH[e_i]$ is a multiple of the *i*-th column of *H*, we know that if there is at most one error in *r* and *Hr* is a multiple *s* of the *i*-th column of *H*, that the error is in the *i*-th position in *r* and the scalar of the error is *s*. The codeword can now be recovered, $c = r - se_i$. Note that the block form of $H = [P \quad I]$ and $G = \begin{bmatrix} I \\ -P \end{bmatrix}$ help to demonstrate how the encoding and error detection work; however, these block forms are not necessarily ideal for the error correction. In practice, we want to choose an ordering which will make it easy to compute *s* and e_i from sHe_i.

5. Conclusions

The Reed-Muller codes have played an important role in the application of coding theory; indeed, they have been used to encode pictures sent from satellites back to Earth. The aim of the Mariner 9 mission in 1971 was to flyover Mars and photograph its entire surface. The pictures had to be transmitted to Earth and, obviously, during this transmission, a lot of errors occurred. The data, therefore, had to be encoded by a very good code; otherwise, all the details which had been detected with the extremely good optical equipment would have remained invisible to us. The pictures had a high resolution of 700x832 pixels. Each pixel became an 8tuple that represented a grey value. These binary data were divided into blocks of 6 bits each; each block was encoded by a codeword of weight 32; thus one paid the price of 26 redundant bits in order to correct errors. For this, a first-order Reed-Muller code of length 64 (generated by all hyper-planes of *PG* (6, 2) was used, which is a 7-error correcting code.

During the late 1940s at Bell laboratories, Richard Hamming decided that a better system was needed. He was allowed to use the computer for research over the weekends. He would put together his punch cards during the week and submit them to be run over the weekend. This would work great as long as his punch cards were completely error-free; however, a single error would cause the computer to pass the job over and move on to the next. He would have to make corrections and resubmit his program at a later time. Richard Hamming thought
that if the computer was smart enough to know that there was a mistake, why not have the computer find the mistake, correct it and continue running the program. He then created the first error correction code, the Hamming Code.

6. References

[1]. D.R. Hughes. Projective Planes. Springer, Berlin-Heidelberg-New York, 1973.

[2]. E.F. Assmus. T.D. Key. Designs and their Codes. Cambridge University Press, Cambridge Tracts in Math. 103, 1992.

[3]. J.H. Van Lint. Introduction to Coding Theory. Springer-Verlag, Berlin-Heidelberg New York, 1992.

[4]. M. Berger. Geometry I. II: Springer-Verlag, Berlin, Heidelberg, 1987

[5]. R. Baer. Projectivities with fixed points on every line of the plane. Bull. Amer. Math. Soc. 52. 273-286, 1946.

[6]. T. Bu. Partitions of a vector space. Discrete Math., vol.31, pp.79–83, Jan. 1980.