Table of Contents

By Square

Working in the **square centric world** of the board implies using a square index of one particular pawn, likely from bitboard traversal.

For doubled (twins) or multiple pawns on a file, we simply intersect the file-mask by own-pawns and perform a population count. This symmetrical property appears to be equal for white and black pawns, except for the before- and behind semantic.

```
wPawnsOnFile = arrFiles[sqOfWhitePawn] & pawnBB[white];
if ( wPawnsOnFile & (wPawnsOnFile-1) ) ->
 white pawn is a least double pawn
if ( wPawnsOnFile & arrNortRay[sqOfWhitePawns] ) ->
 white pawn has own pawn ahead
if ( wPawnsOnFile & arrSoutRay[sqOfWhitePawns] ) ->
 white pawn has own pawn behind

arrFiles[d*]
. . . 1 . . . .
. . . 1 . . . .
. . . 1 . . . .
. . . 1 . . . .
. . . 1 . . . .
. . . 1 . . . .
. . . 1 . . . .
. . . 1 . . . .
```

The arrFiles might be condensed to eight bitboards only, instead of 64 - if we index by

```
file(sq) == sq & 7
```

instead of square.

Multiple Pawns

Working in the **bitboard centric world** to determine pawn related pattern set-wise.

All pawns that are member of the rearspan of all own pawns, have at least one pawn in front on the same file. All pawns that are member of the frontspan of all own pawns, have at least one pawn behind on the same file.

```
// pawns with at least one pawn in front on the same file
U64 wPawnsBehindOwn(U64 wpawns) {return wpawns & wRearSpans(wpawns);}

// pawns with at least one pawn behind on the same file
U64 wPawnsInfrontOwn (U64 wpawns) {return wpawns &
 wFrontSpans(wpawns);}
```

Obviously, since each pawn in front of a pawn implies this pawn behind, the number of pawns in front is equal the number of pawns behind.

```
popCount(wPawnsBehindOwn) == popCount(wPawnsInfrontOwn)
```

The intersection of both sets implies at least a triple, thus a pawn in front and behind.

```
U64 wPawnsInfrontAndBehindOwn (U64 wpawns) {
   return wPawnsInfrontOwn(wpawns) &  wPawnsBehindOwn(wpawns);
}
```

If the intersection is empty on a file, it is a double pawn or a twin.

```
fileWithAtLeastTriple = fileFill(infrontAndBehindOwn);
rearTwin  = behindOwn  & ~fileWithAtLeastTriple;
frontTwin = infrontOwn & ~fileWithAtLeastTriple;
```

Forum Posts

- Doubled and Backward Pawn Engine "Definitions" by Brian Richardson, CCC, September 07, 2009
- Doubled pawns by Stefano Gemma, CCC, May 11, 2016

What links here?

Page	Date Edited
Aegon 1993	Mar 26, 2017
Dispersion and Distortion	Nov 11, 2017
Double and Triple (Bitboards)	May 12, 2016
Doubled Pawn	Mar 27, 2017
Hans Kmoch	Nov 11, 2017
Pawn Pattern and Properties	Nov 11, 2017

Up one Level