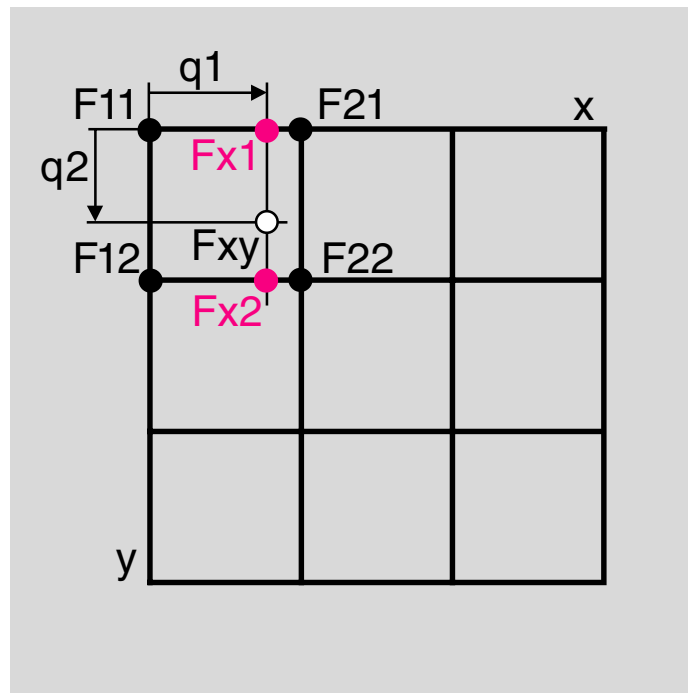


Gernot Hoffmann

# Multidimensional Linear Interpolation



## Contents

1	Linear Interpolation	2
2	Example	3

# 1. Linear Interpolation

Data  $F(x,y,z,u)$  are given in an equally spaced  $n$ -dimensional grid, here for  $n=1$  to  $n=4$  axes. For  $m$  outputs  $F,G,H...$  the interpolation has to be applied  $m$  times.

Entries are given as float numbers  $x,y,z,u$

Next lower integer grid addresses  $x_1,y_1,z_1,u_1$

Next higher integer grid addresses  $x_2,y_2,z_2,u_2$

$$F_{xyz_u} = F(x,y,z,u)$$

$$F_{ijkl} = F(x_i,y_j,z_k,u_l)$$

$$q_1 = (x - x_1) / (x_2 - x_1)$$

$$q_2 = (y - y_1) / (y_2 - y_1)$$

$$q_3 = (z - z_1) / (z_2 - z_1)$$

$$q_4 = (u - u_1) / (u_2 - u_1)$$

$$p_1 = 1 - q_1$$

$$p_2 = 1 - q_2$$

$$p_3 = 1 - q_3$$

$$p_4 = 1 - q_4$$

The program example shows the memory addresses arranged in a binary pattern.

$n = 1$

$$F_x = p_1 F_1 + q_1 F_2$$

$n = 2$

$$F_{x1} = p_1 F_{11} + q_1 F_{21}$$

$$F_{x2} = p_1 F_{12} + q_1 F_{22}$$

$$F_{xy} = p_2 F_{x1} + q_2 F_{x2}$$

$n = 3$

$$F_{x11} = p_1 F_{111} + q_1 F_{211}$$

$$F_{x21} = p_1 F_{121} + q_1 F_{221}$$

$$F_{xy1} = p_2 F_{x11} + q_2 F_{x21}$$

$$F_{x12} = p_1 F_{112} + q_1 F_{212}$$

$$F_{x22} = p_1 F_{122} + q_1 F_{222}$$

$$F_{xy2} = p_2 F_{x12} + q_2 F_{x22}$$

$$F_{xyz} = p_3 F_{xy1} + q_3 F_{xy2}$$

$n = 4$

$$F_{x111} = p_1 F_{1111} + q_1 F_{2111}$$

$$F_{x211} = p_1 F_{1211} + q_1 F_{2211}$$

$$F_{x121} = p_1 F_{1121} + q_1 F_{2121}$$

$$F_{x221} = p_1 F_{1221} + q_1 F_{2221}$$

$$F_{xy11} = p_2 F_{x111} + q_2 F_{x211}$$

$$F_{xy21} = p_2 F_{x121} + q_2 F_{x221}$$

$$F_{xyz1} = p_3 F_{xy11} + q_3 F_{xy21}$$

$$F_{x112} = p_1 F_{1112} + q_1 F_{2112}$$

$$F_{x212} = p_1 F_{1212} + q_1 F_{2212}$$

$$F_{x122} = p_1 F_{1122} + q_1 F_{2122}$$

$$F_{x222} = p_1 F_{1222} + q_1 F_{2222}$$

$$F_{xy12} = p_2 F_{x112} + q_2 F_{x212}$$

$$F_{xy22} = p_2 F_{x122} + q_2 F_{x222}$$

$$F_{xyz2} = p_3 F_{xy12} + q_3 F_{xy22}$$

$$F_{xyz_u} = p_4 F_{xyz1} + q_4 F_{xyz2}$$

## 2.1 Example

This example shows the interpolation for applications which we encounter in ICC profiles. We convert from RGB to CMYK by a Look-Up Table with 3 inputs and 4 outputs, and from CMYK to RGB by a LUT with 4 inputs and 3 outputs.

A simple conversion model applies additionally Under Color Removal (UCR).

Actually, in ICC profiles the conversion is never done directly between RGB and CMYK. It's just a tutorial example - structurally relevant, not literally.

```
Program ZMultipl;
{   Project:           Multidimensional Interpolation
  Author:             G.Hoffmann
  Date, last revision: ref. Rdate

{ Simulate Linear Interpolation in 3- and 4-dimensional
  Look-up tables with 4 and 3 outputs

{$A+,B-,D-,E-,F-,G+,I+,L+,N+,O-,P-,Q-,R-,S-,T-,V-,X-,Y-}
{$C Moveable PreLoad Permanent }

Uses      Crt,Dos,Zefir30,Zefir31;
Const     Rname='ZMultipl';
          RDate='10.06.2003';
          gp=8; { grid points 0,1,...,gp-1 }
          g1=gp-1; g2=gp*gp; g3=gp*g2; g4=g2*g2;
Type      Trgb =Record r,g,b   : Single; End;
          Tcmyk=Record c,m,y,k : Single; End;
Var       rgb: Trgb;
          cmyk: Tcmyk;
          FRC: Array[0..g3-1] Of Tcmyk;
          FCR: Array[0..g4-1] Of Trgb;

Procedure FillFRC(UCR: Boolean);
{ RGB to CMYK, simple conversion with UCR }
Var rr,gg,bb,p   : Integer;
    r,g,b,dc,fc  : Single;
Begin
dc:=255/g1; fc:=100/255; p:=0;
r:=0;
For rr:=0 to g1 Do
  Begin
  g:=0;
  For gg:=0 to g1 Do
    Begin
    b:=0;
    For bb:=0 to g1 Do
      Begin
      With FRC[p] Do
        Begin
          c:=(255-r)*fc; m:=(255-g)*fc; y:=(255-b)*fc; k:=0;
          If UCR Then
            Begin
              k:=c; If m<k Then k:=m; If y<k Then k:=y;
              c:=c-k; m:=m-k; y:=y-k;
            End;
          End;
          Inc(p);
          b:=b+dc;
          End;
          g:=g+dc;
          End;
          r:=r+dc;
          End;
        End;
      End;
    End;
  End;
End;
```

## 2.2 Example

```
Procedure FillFCR(UCR: Boolean);
{ CMYK to RGB }
Var cc,mm,yy,kk,p: Integer;
    c,m,y,k,dc,fc: Single;
Begin
dc:=100/g1; fc:=255/100; p:=0;
c:=0;
For cc:=0 to g1 Do
  Begin
  m:=0;
  For mm:=0 to g1 Do
    Begin
    y:=0;
    For yy:=0 to g1 Do
      Begin
      k:=0;
      For kk:=0 to g1 Do
        Begin
        With FCR[p] Do
          Begin
          If UCR Then
            r:=(100-c-k)*fc; g:=(100-m-k)*fc; b:=(100-y-k)*fc
          End Else
            Begin
            r:=(100-c)*fc; g:=(100-m)*fc; b:=(100-y)*fc ;
            End;
          End;
          Inc(p);
          k:=k+dc;
        End;
      y:=y+dc;
      End;
    m:=m+dc;
    End;
  c:=c+dc;
  End;
End;
```

```
Procedure RGBtoCMYK(rgb: Trgb; Var cmyk: Tcmyk);
Var i1,j1,k1,i2,j2,k2 : Integer;
    fc,q1,q2,q3,p1,p2,p3 : Single;
Function P(i,j,k: Integer): Integer;
  Begin
  P:=(i*gp+j)*gp+k;
  End;
Begin
fc:=g1/255;
With rgb Do
  Begin
  If r<0 Then r:=0; If r>255 Then r:=255;
  If g<0 Then g:=0; If g>255 Then g:=255;
  If b<0 Then b:=0; If b>255 Then b:=255;
  i1:=Trunc(r*fc); If i1<0 Then i1:=0;
  i2:=i1+1; If i2>g1 Then i2:=g1;
  j1:=Trunc(g*fc); If j1<0 Then j1:=0;
  j2:=j1+1; If j2>g1 Then j2:=g1;
  k1:=Trunc(b*fc); If k1<0 Then k1:=0;
  k2:=k1+1; If k2>g1 Then k2:=g1;
  q1:=r*fc-i1; q2:=g*fc-j1; q3:=b*fc-k1;
  p1:=1-q1; p2:=1-q2; p3:=1-q3;
  End;
cmyk.c:=p3*(p2*( p1*FRC[P(i1,j1,k1)].c +
  q1*FRC[P(i2,j1,k1)].c) +
  q2*( p1*FRC[P(i1,j2,k1)].c +
  q1*FRC[P(i2,j2,k1)].c))+
  q3*(p2*( p1*FRC[P(i1,j1,k2)].c +
  q1*FRC[P(i2,j1,k2)].c) +
  q2*( p1*FRC[P(i1,j2,k2)].c +
  q1*FRC[P(i2,j2,k2)].c));
```

## 2.3 Example

```

cmyk.m:=p3*(p2*( p1*FRC[P(i1,j1,k1)].m  +
                q1*FRC[P(i2,j1,k1)].m)  +
            q2*( p1*FRC[P(i1,j2,k1)].m  +
                q1*FRC[P(i2,j2,k1)].m))+
q3*(p2*( p1*FRC[P(i1,j1,k2)].m  +
        q1*FRC[P(i2,j1,k2)].m)  +
    q2*( p1*FRC[P(i1,j2,k2)].m  +
        q1*FRC[P(i2,j2,k2)].m));
cmyk.y:=p3*(p2*( p1*FRC[P(i1,j1,k1)].y  +
                q1*FRC[P(i2,j1,k1)].y)  +
            q2*( p1*FRC[P(i1,j2,k1)].y  +
                q1*FRC[P(i2,j2,k1)].y))+
q3*(p2*( p1*FRC[P(i1,j1,k2)].y  +
        q1*FRC[P(i2,j1,k2)].y)  +
    q2*( p1*FRC[P(i1,j2,k2)].y  +
        q1*FRC[P(i2,j2,k2)].y));
cmyk.k:=p3*(p2*( p1*FRC[P(i1,j1,k1)].k  +
                q1*FRC[P(i2,j1,k1)].k)  +
            q2*( p1*FRC[P(i1,j2,k1)].k  +
                q1*FRC[P(i2,j2,k1)].k))+
q3*(p2*( p1*FRC[P(i1,j1,k2)].k  +
        q1*FRC[P(i2,j1,k2)].k)  +
    q2*( p1*FRC[P(i1,j2,k2)].k  +
        q1*FRC[P(i2,j2,k2)].k));

End;

Procedure CMYKtoRGB(cmyk: Tcmyk; Var rgb: Trgb);
Var i1,j1,k1,i2,j2,k2,l1,l2 : Integer;
    fc,q1,q2,q3,q4,p1,p2,p3,p4: Single;
Function P(i,j,k,l: Integer): Integer;
Begin
P:=(i*gp+j)*gp+k)*gp+l;
End;
Begin
fc:=g1/100;
With cmyk Do
Begin
If c<0 Then c:=0;      If c>100 Then c:=100;
If m<0 Then m:=0;      If m>100 Then m:=100;
If y<0 Then y:=0;      If y>100 Then y:=100;
If k<0 Then k:=0;      If k>100 Then k:=100;
i1:=Trunc(c*fc);      If i1<0 Then i1:=0;
i2:=i1+1;              If i2>g1 Then i2:=g1;
j1:=Trunc(m*fc);      If j1<0 Then j1:=0;
j2:=j1+1;              If j2>g1 Then j2:=g1;
k1:=Trunc(y*fc);      If k1<0 Then k1:=0;
k2:=k1+1;              If k2>g1 Then k2:=g1;
l1:=Trunc(k*fc);      If l1<0 Then l1:=0;
l2:=l1+1;              If l2>g1 Then l2:=g1;
q1:=c*fc-i1; q2:=m*fc-j1; q3:=y*fc-k1; q4:=k*fc-l1;
p1:=1-q1; p2:=1-q2; p3:=1-q3; p4:=1-q4;
End;
rgb.r:=p4*(p3*(p2*( p1*FCR[P(i1,j1,k1,l1)].r  +
                    q1*FCR[P(i2,j1,k1,l1)].r)  +
                q2*( p1*FCR[P(i1,j2,k1,l1)].r  +
                    q1*FCR[P(i2,j2,k1,l1)].r))+
            q3*(p2*( p1*FCR[P(i1,j1,k2,l1)].r  +
                    q1*FCR[P(i2,j1,k2,l1)].r)  +
                q2*( p1*FCR[P(i1,j2,k2,l1)].r  +
                    q1*FCR[P(i2,j2,k2,l1)].r))+
            q4*(p3*(p2*( p1*FCR[P(i1,j1,k1,l2)].r  +
                    q1*FCR[P(i2,j1,k1,l2)].r)  +
                q2*( p1*FCR[P(i1,j2,k1,l2)].r  +
                    q1*FCR[P(i2,j2,k1,l2)].r))+
            q3*(p2*( p1*FCR[P(i1,j1,k2,l2)].r  +
                    q1*FCR[P(i2,j1,k2,l2)].r)  +
                q2*( p1*FCR[P(i1,j2,k2,l2)].r  +
                    q1*FCR[P(i2,j2,k2,l2)].r)));

```

## 2.4 Example

```
rgb.g:=p4*(p3*(p2*( p1*FCR[P(i1,j1,k1,l1)].g +
                    q1*FCR[P(i2,j1,k1,l1)].g) +
                    q2*( p1*FCR[P(i1,j2,k1,l1)].g +
                    q1*FCR[P(i2,j2,k1,l1)].g)) +
                    q3*(p2*( p1*FCR[P(i1,j1,k2,l1)].g +
                    q1*FCR[P(i2,j1,k2,l1)].g) +
                    q2*( p1*FCR[P(i1,j2,k2,l1)].g +
                    q1*FCR[P(i2,j2,k2,l1)].g)))+
                    q4*(p3*(p2*( p1*FCR[P(i1,j1,k1,l2)].g +
                    q1*FCR[P(i2,j1,k1,l2)].g) +
                    q2*( p1*FCR[P(i1,j2,k1,l2)].g +
                    q1*FCR[P(i2,j2,k1,l2)].g)) +
                    q3*(p2*( p1*FCR[P(i1,j1,k2,l2)].g +
                    q1*FCR[P(i2,j1,k2,l2)].g) +
                    q2*( p1*FCR[P(i1,j2,k2,l2)].g +
                    q1*FCR[P(i2,j2,k2,l2)].g))));
rgb.b:=p4*(p3*(p2*( p1*FCR[P(i1,j1,k1,l1)].b +
                    q1*FCR[P(i2,j1,k1,l1)].b) +
                    q2*( p1*FCR[P(i1,j2,k1,l1)].b +
                    q1*FCR[P(i2,j2,k1,l1)].b)) +
                    q3*(p2*( p1*FCR[P(i1,j1,k2,l1)].b +
                    q1*FCR[P(i2,j1,k2,l1)].b) +
                    q2*( p1*FCR[P(i1,j2,k2,l1)].b +
                    q1*FCR[P(i2,j2,k2,l1)].b)))+
                    q4*(p3*(p2*( p1*FCR[P(i1,j1,k1,l2)].b +
                    q1*FCR[P(i2,j1,k1,l2)].b) +
                    q2*( p1*FCR[P(i1,j2,k1,l2)].b +
                    q1*FCR[P(i2,j2,k1,l2)].b)) +
                    q3*(p2*( p1*FCR[P(i1,j1,k2,l2)].b +
                    q1*FCR[P(i2,j1,k2,l2)].b) +
                    q2*( p1*FCR[P(i1,j2,k2,l2)].b +
                    q1*FCR[P(i2,j2,k2,l2)].b))));
End;

Procedure Show(UCR: Boolean);
Var i      : Integer;
    rs,gs,bs : Single;
Begin
Randseed:=1;
FillFRC(UCR);
FillFCR(UCR);
For i:=1 to 5 Do
Begin
With rgb Do
Begin
r:=random(255); g:=random(255); b:=random(255); rs:=r; gs:=g; bs:=b;
End;
RGBtoCMYK(rgb,cmyk);
With cmyk Do
Begin Writeln(rs:10:4,gs:10:4,bs:10:4,c:10:4,m:10:4,y:10:4,k:10:4); End;
CMYKtoRGB(cmyk,rgb);
With rgb Do
Begin Writeln(rs:10:4,gs:10:4,bs:10:4,r:10:4,g:10:4,b:10:4); End;
End;
Writeln;
End;

BEGIN
ClrScr;
Show(False);
Show(True);
Stop;
END.
```