```
function [ I ] = bewegung()
%Einlesen der Bilder
I1 = double(imread('bilder\bewegung1.bmp', 'BMP'));
I2 = double(imread('bilder\bewegung2.bmp', 'BMP'));

%Berechnung der Dimension
D = size(I1);
n = D(1,1);
m = D(1,2);

% 2.1 Randbedingungen Erweitere die Matrix 1.

I11 = I1(1,:);
I1 = [I11;I1];
I21 = I1(end,:);
I1 = [I1;I21];
I31 = I1(:,1);
I1 = [I31,I1];
I41 = I1(:,end);
I1 = [I1,I41];
% 2.1 Randbedingungen Erweitere die Matrix 2.
I12 = I2(1,:);
I2 = [I12;I2];
I22 = I2(end,:);
I2 = [I2;I22];
I32 = I2(:,1);
I2 = [I32,I2];
I42 = I2(:,end);
I2 = [I2,I42];

%Berechnung von Ix Iy und It
for i = 2:n+1;
for j = 2:m+1;
    Ix(i-1,j-1) = (1/4 *
    (I1(i+1,j)-I1(i,j)+I1(i+1,j+1)-I1(i,j+1)+I2(i+1,j)-I2(i,j)+I2(i+1,j+1)-I2(i,j+1)));
    Iy(i-1,j-1) = (1/4 *
    (I1(i,j+1)-I1(i,j)+I1(i+1,j+1)-I1(i+1,j)+I2(i,j+1)-I2(i,j)+I2(i+1,j+1)-I2(i+1,j)));
    It(i-1,j-1) = (1/4 *
    (I2(i,j+1)-I1(i,j+1)+I2(i+1,j+1)-I1(i+1,j+1)+I2(i+1,j)-I1(i+1,j)+I2(i,j)-I1(i,j)));

end
end
%Erweitere Matrix um 2 Stellen je Seite;
Ix = [ones(2,100);Ix;ones(2,100)];
Ix = [ones(104,2),Ix,ones(104,2)];

Iy = [ones(2,100);Iy;ones(2,100)];
Iy = [ones(104,2),Iy,ones(104,2)];

It = [ones(2,100);It;ones(2,100)];
It = [ones(104,2),It,ones(104,2)];


%Erstelle nun Ausgleichsproblem mit Lösung.
for i = 3:n+2;
for j = 3:m+2;
```

```
   a1 = Ix(i-2,j-2); b1 = Ix(i-2,j-1); c1 = Ix(i-2,j); d1 = Ix(i-2,j+1); e1 = Ix(i-2,j+2);
   a2 = Ix(i-1,j-2); b2 = Ix(i-1,j-1); c2 = Ix(i-1,j); d2 = Ix(i-1,j+1); e2 = Ix(i-1,j+2);
   a3 = Ix(i,j-2);   b3 = Ix(i,j-1);   c3 = Ix(i,j);   d3 = Ix(i,j+1);   e3 = Ix(i,j+2);
   a4 = Ix(i+1,j-2); b4 = Ix(i+1,j-1); c4 = Ix(i+1,j); d4 = Ix(i+1,j+1); e4 = Ix(i+1,j+2);
   a5 = Ix(i+2,j-2); b5 = Ix(i+2,j-1); c5 = Ix(i+2,j); d5 = Ix(i+2,j+1); e5 = Ix(i+2,j+2);

  Px = [a1 b1 c1 d1 e1 a2 b2 c2 d2 e2 a3 b3 c3 d3 e3 a4 b4 c4 d4 e4 a5 b5 c5 d5 e5];

   a1 = Iy(i-2,j-2); b1 = Iy(i-2,j-1); c1 = Iy(i-2,j); d1 = Iy(i-2,j+1); e1 = Iy(i-2,j+2);
   a2 = Iy(i-1,j-2); b2 = Iy(i-1,j-1); c2 = Iy(i-1,j); d2 = Iy(i-1,j+1); e2 = Iy(i-1,j+2);
   a3 = Iy(i,j-2);   b3 = Iy(i,j-1);   c3 = Iy(i,j);   d3 = Iy(i,j+1);   e3 = Iy(i,j+2);
   a4 = Iy(i+1,j-2); b4 = Iy(i+1,j-1); c4 = Iy(i+1,j); d4 = Iy(i+1,j+1); e4 = Iy(i+1,j+2);
   a5 = Iy(i+2,j-2); b5 = Iy(i+2,j-1); c5 = Iy(i+2,j); d5 = Iy(i+2,j+1); e5 = Iy(i+2,j+2);

   Py = [a1 b1 c1 d1 e1 a2 b2 c2 d2 e2 a3 b3 c3 d3 e3 a4 b4 c4 d4 e4 a5 b5 c5 d5 e5];

   A = [Px' , Py'];

   a1 = It(i-2,j-2); b1 = It(i-2,j-1); c1 = It(i-2,j); d1 = It(i-2,j+1); e1 = It(i-2,j+2);
   a2 = It(i-1,j-2); b2 = It(i-1,j-1); c2 = It(i-1,j); d2 = It(i-1,j+1); e2 = It(i-1,j+2);
   a3 = It(i,j-2);   b3 = It(i,j-1);   c3 = It(i,j);   d3 = It(i,j+1);   e3 = It(i,j+2);
   a4 = It(i+1,j-2); b4 = It(i+1,j-1); c4 = It(i+1,j); d4 = It(i+1,j+1); e4 = It(i+1,j+2);
   a5 = It(i+2,j-2); b5 = It(i+2,j-1); c5 = It(i+2,j); d5 = It(i+2,j+1); e5 = It(i+2,j+2);

   b = -1 * [a1 b1 c1 d1 e1 a2 b2 c2 d2 e2 a3 b3 c3 d3 e3 a4 b4 c4 d4 e4 a5 b5 c5 d5 e5]';
   %Berechne x Stern
   M = pinv(A);
   %xs ist der Vektor mit u und v
   xs =  M * b;
    u(i-2,j-2) = xs(1,1);
    v(i-2,j-2) = xs(2,1);

   I(i-2,j-2) = norm(((A * xs) - b),2);

end
end


quiver(v(1:10:end,1:10:end),u(1:10:end,1:10:end))
end
```