0.1 Isotropic point normalization: `pointnorm`

Normalization of point coordinates in order to achieve better numerical stability of direct linear transform (DLT) estimation. See [?] for more details.

function `[u2,T] = pointnorm(u)`

input

u	[3×N]	Matrix of unnormalized coordinates of N points.

output

u2	[3×N]	Normalized coordinates.
T	[3×3]	Transformation matrix, `u2 = Tu`.

see also `u2Fdlt` (p. ??), `u2Hdlt` (p. ??).

Center the coordinates.

```
centroid = mean( u(1:2,:)' )';
u2 = u;
u2(1:2,:) = u(1:2,:) - repmat(centroid,1,n);
```

Scale points to have average distance from the origin $\sqrt{2}$.

```
scale = sqrt(2) / mean( sqrt(sum(u2(1:2,:).^2)) );
u2(1:2,:) = scale*u2(1:2,:);
```

Composition of the normalization matrix.

```
T = diag([scale scale 1]);
T(1:2,3) = -scale*centroid;
```

Example of `pointnorm` usage

The function `pointnorm` is used as follows. Generate and display artificial 2D data:

```
u = 100*rand(2,100);
u(3,:) = 1; % make the data homogeneous
figure(1);  clf
plot( u(1,:), u(2,:), '+' ); hold on
title('original␣points')
```

Normalize points such that centroid of $\mathbf{u}_2$ will be $[0,0]^\top \mathbf{u}_2 = \mathtt{T}\,\mathbf{u}_1$:

```
[u2,T] = pointnorm(u);
```

Control computation:

```
u3 = inv(T)*u2;
figure(1)
plot( u3(1,:), u3(2,:), 'ko', 'MarkerSize',10 )
```

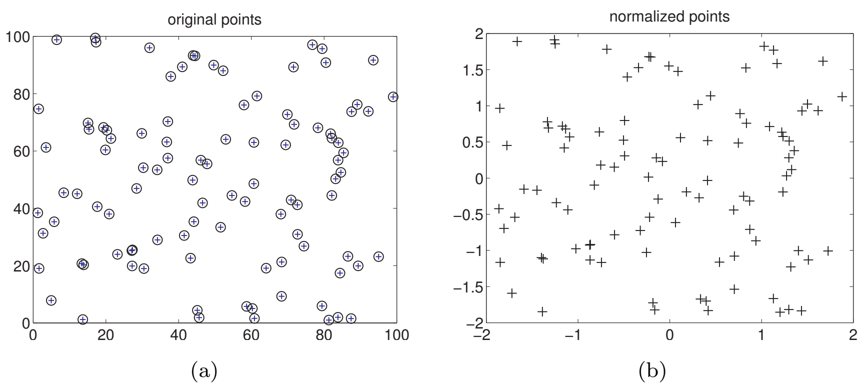

Figure 1: (a) Original points `u` (crosses) and re-computed points circles `u3`. (b) Normalized coordinates `u2`.