warp

keys:	**[from]**	*<number>*	source picture
	[to]	*<number>*	output picture
	image	*<number>*	*Plist* containing image control points
	map	*<number>*	*Plist* containing map control points
	order	*<number>*	order of mapping polynomial
	layer	*<number>*	resample specified picture layer
	position	*<x>,<y>*	position in map image space of the output image
	size	*<x>, <y>*	size of output picture
	stride	*<number>*	scale factor in output image
	errmax	*<number>*	maximum error of control points
options:	**bilinear/bicubic**		resampling method
	verify		verify information about geometric correction at the console

The **warp** command corrects geometrical distortions in images by fitting a polynomial grid to a set of control points. This command can be used in *Remote Sensing* applications.

Examples

```
warp 2 3 map 4 image 5
```

This command resamples picture 2 to picture 3 using map control points in picture 4 and image control points in picture 5.

```
warp 2 3 map 4 image 5 bicubic order 2
```

This command performs the same functions as the above command but uses a second order polynomial and resamples using bicubic resampling.

```
warp 2 3 map 4 image 5 position 120,100 size 64 stride 10
```

This command resamples image 2 to image 3 using *nearest neighbour* interpolation. A first order polynomial is used with pictures 4 and 5 containing the control points. The area covered by the output image is:

(120.0, 100.0) .. (760.0, 740.0)

since (64 * 10.0 = 640.0)

warp

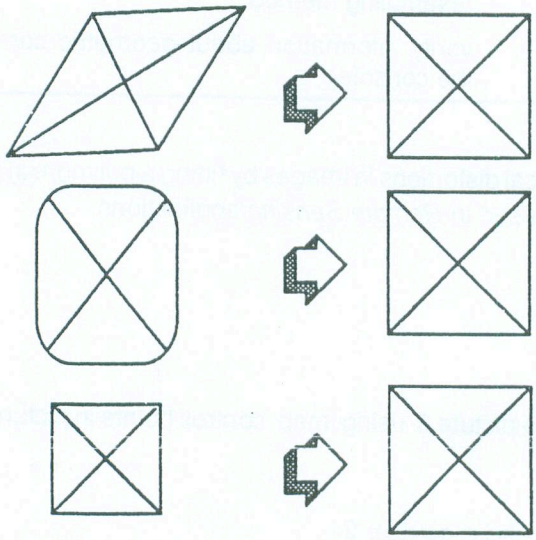
Description

The **warp** command warps the source picture using a polynomial, the coefficients of which are calculated from the control points using the least squares method. The maximum order of polynomial is at least 3 but may be more, depending on the length of a Semper row buffer. You supply the control points in a *Plist* picture, specified by the **map** or **image** keys. You can specify three methods of interpolation:

- *nearest neighbour* (the default)
- **bilinear** interpolation
- **bicubic** interpolation

Use the **layer** key to specify a layer of the picture to resample.

The diagram below illustrates some uses of the **warp** command:



By default, the output picture is the same size as the input image. To change the dimensions, use the **size** key. Use the **position** key to change the position of the output image in map (reference image) space. The point specified by **position** is the bottom left of the image. Use the **stride** key to define a scale factor for the image. Note that if **stride** is greater than 1 then for similarly scaled images, features will shrink.

Use the **verify** option to list the registration accuracy at the console. This consists of the control points and their calculated position as compared with their given position. The (standard) error in the *x* and *y* directions is also listed.

Use the **errmax** key to define the maximum displacement that is allowed between a control point and its calculated position. If **warp** finds a point that exceeds this value then the point is dropped and

warp

the coefficients are re-calculated. If not enough 'good' points remain the command gives an error message. This process is repeated until all points are within the required accuracy or not enough points remain. If you specify the **verify** option, the accuracy information is displayed for each cycle.

The first order polynomial is as follows:

$$u = a0 + a1.x + a2.y$$
$$v = b0 + b1.x + b2.y$$

The second order polynomial is given below:

$$u = a0 + a1.x + a2.y + a3.x^2 + a4.x.y + a5.y^2$$
$$v = b0 + b1.x + b2.y + b3.x^2 + b4.x.y + b5.y^2$$

where *u, v* refer to uncorrected image coordinates and *x, y* refer to map reference coordinates.

Defaults and Ranges

keys/options	defaults	range
[from]	current picture, held in the variable *select*	valid picture number
[to]	source picture	valid picture number
image	*none*	valid *Plist* picture number
map	*none*	valid *Plist* picture number
order	1	positive integer
layer	all layers	integer in range 1 to number of layers
position	position 0,0	positive integers
size	size of input image	positive integers
stride	1	positive integer
errmax	*none*	positive integer
bilinear/bicubic	*nearest neighbour*	
verify	verification off	