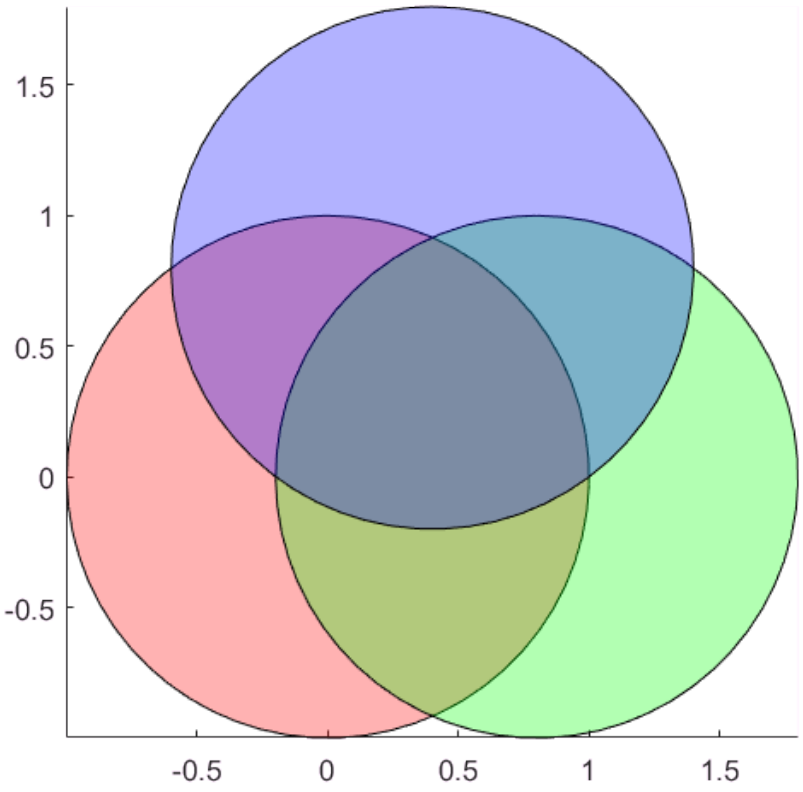
Changing Transparency of Images, Patches or Surfaces

This example shows how to modify transparency of images, patches and surfaces.

Transparency for All Objects in Axes

Transparency values are referred to as alpha values. Use the `alpha` function to set the transparency for all image, patch, and surface objects in the current axes. Specify a transparency value between 0 (fully transparent) and 1 (fully opaque).

```
t = 0:0.1:2*pi;
x = sin(t);
y = cos(t);

figure
patch(x,y,'r')              % make a red circular patch
patch(x+0.8,y,'g')          % make a green circular path
patch(x+0.4,y+0.8,'b')      % make a blue circular path
axis square tight           % set axis to square

alpha(0.3)                  % set all patches transparency to 0.3
```



Transparency for Individual Surfaces

The transparency of a surface is defined by its `AlphaData` property. Set the alpha data as either a scalar value or a matrix of values specifying the transparency of each vertex of the surface. The `FaceAlpha` property indicates how the transparency of the surface faces are determined from vertex transparency.

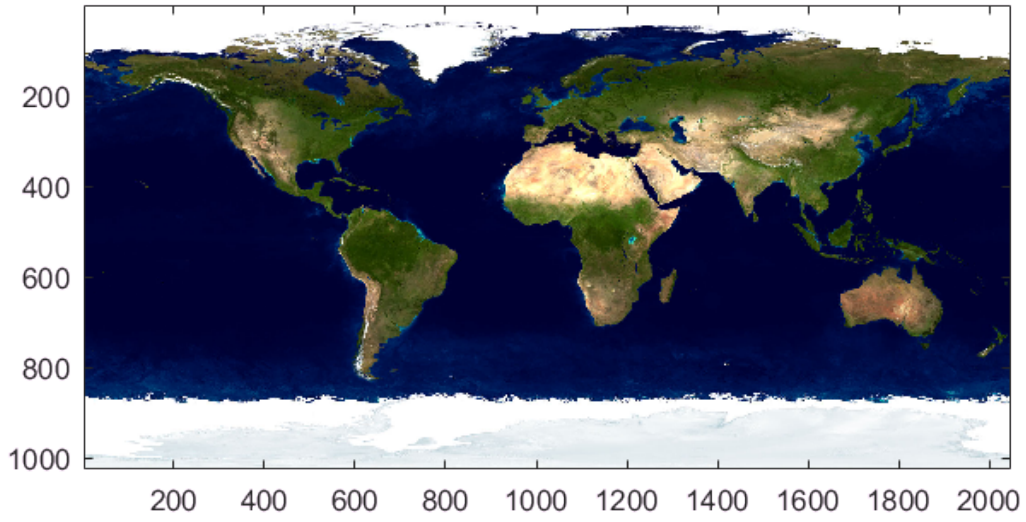
```
[X,Y,Z] = peaks(20);
s2 = surf(X,Y,Z);

s2.AlphaData = gradient(Z);    % set vertex transparencies
s2.FaceAlpha = 'flat';
```

Transparency for Individual Images

Like surfaces, the transparency of an image is also defined by its `AlphaData` property. For images, set the alpha data as either a scalar value or a matrix of values specifying the transparency of each element in the image data.

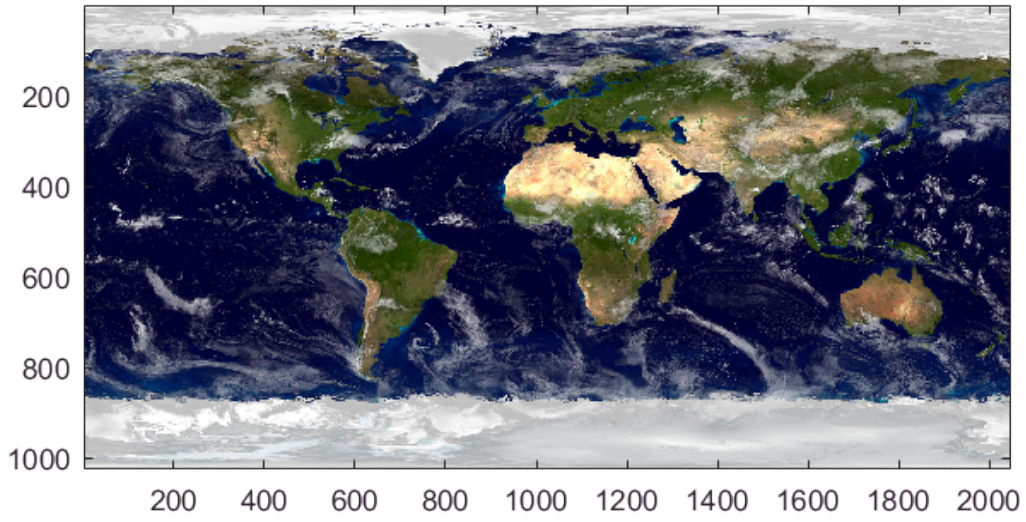
For example, use transparency to overlay two images. First, display the image of the Earth.

```
earth = imread('landOcean.jpg');
image(earth)    % display Earth image
axis image
```



Then, add a cloud layer to the image of the Earth using transparency.

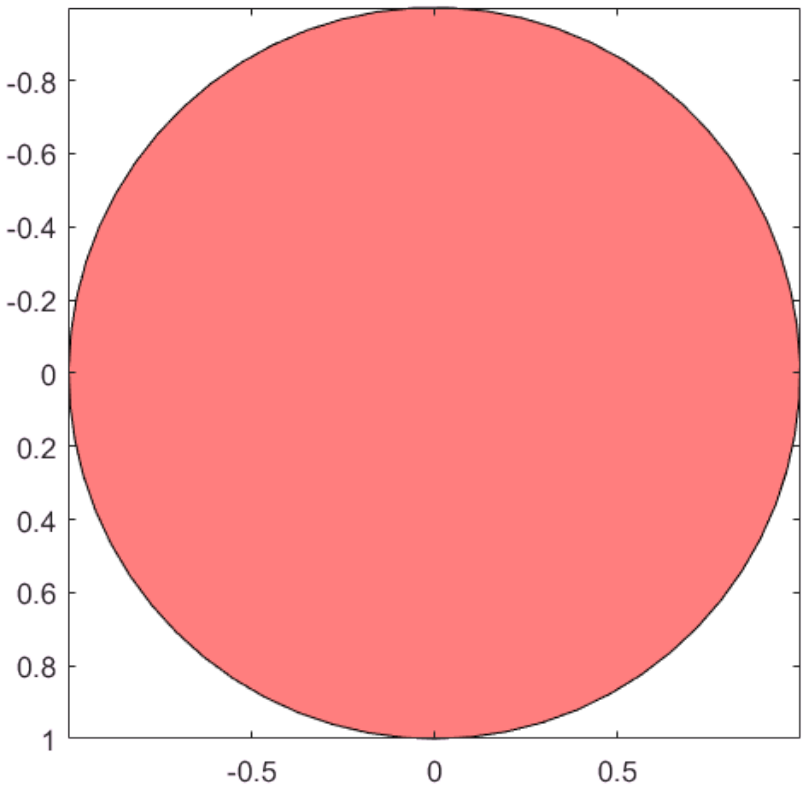
```
clouds = imread('cloudCombined.jpg');
image(earth)
axis image
hold on

im = image(clouds);
im.AlphaData = max(clouds,[],3);    % set transparency to maximum cloud value
hold off
```



Transparency for Individual Patches

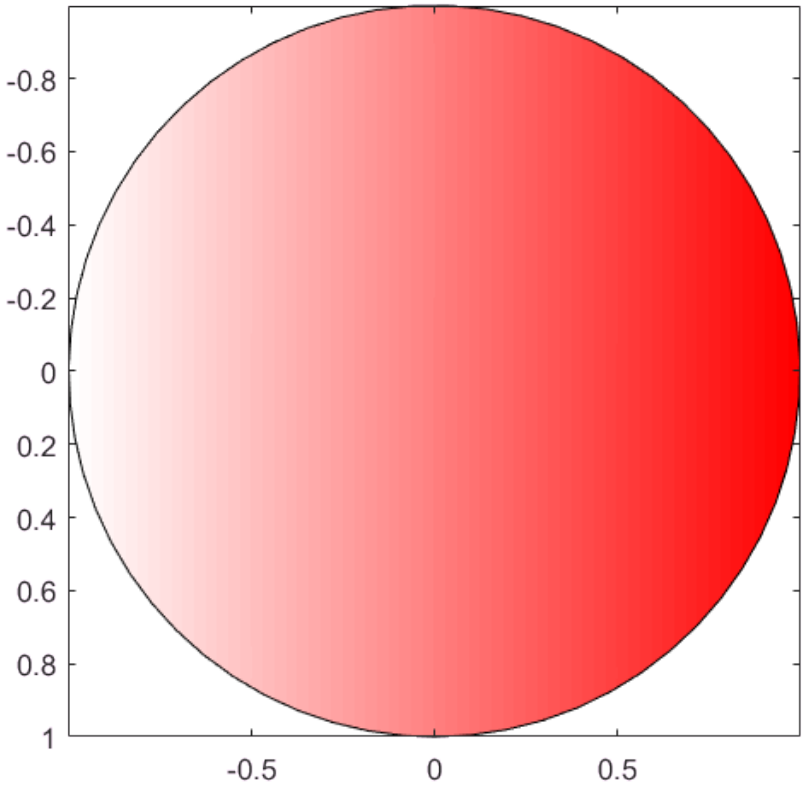
The transparency of a patch is defined by its `FaceAlpha` and `FaceVertexAlphaData` properties. For constant transparency across the entire patch, set the `FaceVertexAlphaData` to a constant between 0 (fully transparent) and 1 (fully opaque), and set the `FaceAlpha` property to `'flat'`.

```
cla
p1 = patch(x,y,'r');              % make a red circular patch
axis square tight                 % set axis to square

p1.FaceVertexAlphaData = 0.2;     % Set constant transparency
p1.FaceAlpha = 'flat' ;           % Interpolate to find face transparency
```



For transparency that varies across the patch, set the `FaceVertexAlphaData` to a matrix of values specifying the transparency at each vertex or each face of the patch. The `FaceAlpha` property then indicates how the face transparencies are determined using the `FaceVertexAlphaData`. If alpha data is specified for vertices, `FaceAlpha` must be set to `'interp'`.

```
p1.FaceVertexAlphaData = x';   % Set vertex transparency to x values
p1.FaceAlpha = 'interp' ;      % Interpolate to find face transparency
```



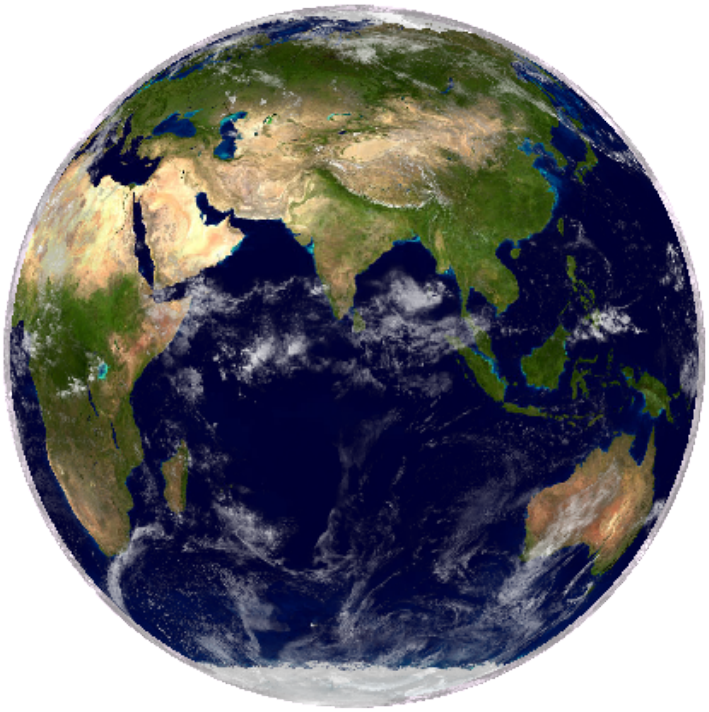
Transparency with Texture Mapping

Texture mapping maps a 2-D image onto a 3-D surface. An image can be mapped to a surface by setting the `CData` property to the image data and setting the `FaceColor` property to be `'texturemap'`.

This example creates a 3-D view of the earth and clouds. It creates spherical surfaces and uses texture mapping to map the images of the earth and clouds onto the surfaces.

```
[px,py,pz] = sphere(50);                % generate coordinates for a 50 x 50 sphere

cla
sEarth = surface(py, px ,flip(pz));
sEarth.FaceColor = 'texturemap';        % set color to texture mapping
sEarth.EdgeColor = 'none';              % remove surface edge color
sEarth.CData = earth;                   % set color data

hold on
sCloud = surface(px*1.02,py*1.02,flip(pz)*1.02);

sCloud.FaceColor = 'texturemap';        % set color to texture mapping
sCloud.EdgeColor = 'none';              % remove surface edge color
sCloud.CData = clouds;                  % set color data

sCloud.FaceAlpha = 'texturemap';        % set transparency to texture mapping
sCloud.AlphaData = max(clouds,[],3);    % set transparency data
hold off
```

```
view([80 2])                            % specify viewpoint
daspect([1 1 1])                        % set aspect ratio
axis off tight                          % remove axis and set limits to data range
```



The images used in this example are from Visible Earth.

Credit: NASA Goddard Space Flight Center Image by Reto Stöckli (land surface, shallow water, clouds). Enhancements by Robert Simmon (ocean color, compositing, 3D globes, animation). Data and technical support: MODIS Land Group; MODIS Science Data Support Team; MODIS Atmosphere Group; MODIS Ocean Group Additional data: USGS EROS Data Center (topography); USGS Terrestrial Remote Sensing Flagstaff Field Center (Antarctica); Defense Meteorological Satellite Program (city lights).