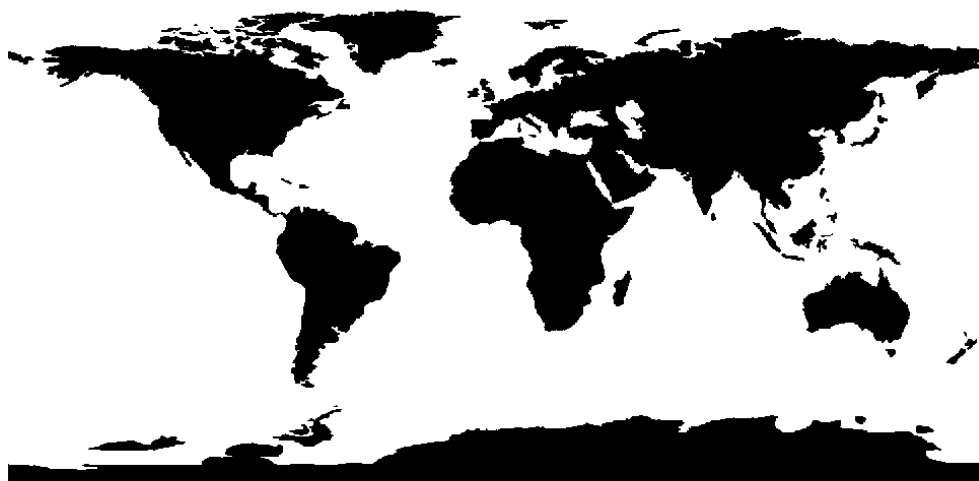

```
close all;
im = imread('map.gif');
gray = mat2gray(im);
[rows, cols] = size(gray);
y = 1:rows;
```

```
figure;
imshow(gray);
```

Warning: Image is too big to fit on screen; displaying at 67%



cylindrical projection

```
cylindrical = zeros(size(gray));
val = [];
```

```
% Transformation
% The rows are first scaled to range from [-360,360] and then converted to
% radians by multiplying by pi/rows. This converts the range of sine from
% 0 to pi/2. The sine transformation is then performed and the transformed
% rows are converted back to the original scale.
```

```
for j = 1:size(gray,1)
    newX = round((sin((j - rows/2)/(rows/pi)) * (rows/pi) ) + (rows/2)) + 1;
    val = [val newX];
    cylindrical(newX,:) = gray(j,:);
end
```

```
%figure;
%imshow(cylindrical);
intp = setdiff(1:size(gray,1), val);
out = cylindrical(unique(val),:);
```

```
figure;  
imshow(out);
```

Warning: Image is too big to fit on screen; displaying at 67%



mercator projection

```
mercator = zeros(size(gray));  
val = [];  
  
% Transformation  
% The rows are first scaled to range from [-360,360] and then converted to  
% radians by multiplying by pi/rows. This converts the range of tan from  
% 0 to pi/2. A relaxation factor of 0.95 is multiplied as tan(pi/2) is  
% infinity. The log transformation is then performed and the transformed rows are  
xnew = log(tan((pi/4)+((y - rows/2)*pi*0.95/rows)/2)) * rows/(pi*0.95);  
  
%scale the new row range to start from 1  
xnew = round(xnew - min(xnew)) + 1;  
  
%assign values to the projected rows  
mercator(xnew,:) = gray;  
  
%assign values to the unassigned rows of projection by average value of the  
%nearest assigned rows  
for i = 2:rows  
    temp = round((gray(i,:) + gray(i-1,:))/2);  
    range = xnew(i-1):xnew(i);  
    mercator(range,:) = repmat(temp,xnew(i)-xnew(i-1)+1,1);  
end  
  
figure;  
imshow(mercator);
```

Warning: Image is too big to fit on screen; displaying at 33%



Published with MATLAB® 8.0