MATLAB code for generating sinusoidal gratings (OMR).

|  |
| --- |
| totalStages = 11;  stageDurations = [30, 30, 30, 30, 30, 30, 30, 30, 30, 30, 30];  pauseDuration = 3;  angularSpeeds = [4, 4, 4, 4, 3, 3, 3, 3, 3, 3, 3];  densities = [0.0125, 0.025, 0.0375, 0.05, 0.065, 0.075, 0.0875, 0.100, 0.1125, 0.125, 0.1375];  imageWidth = 1280;  imageHeight = 1024;  figure('Units', 'pixels',...  'Position', [0 0 imageWidth imageHeight],...  'Color', 'k',...  'MenuBar', 'none',...  'ToolBar', 'none',...  'DoubleBuffer', 'on');  hAx = axes('Position', [0 0 1 1]);  axis off tight;  colormap(gray);  hImage = image(hAx, 'CData', [],...  'XData', [1 imageWidth],...  'YData', [1 imageHeight]);  set(hAx, 'XLim', [1 imageWidth],...  'YLim', [1 imageHeight],...  'YDir', 'reverse',...  'DataAspectRatio', [1 1 1]);  greyBackground = uint8(0.5 \* 255 \* ones(imageHeight, imageWidth));  videoFilename = 'E:\OMR code\MP4';  video = VideoWriter(videoFilename, 'MPEG-4');  video.FrameRate = 30;  video.Quality = 100;  open(video);  initialPauseDuration = 3;  initialPauseFrames = round(initialPauseDuration \* video.FrameRate);  set(hImage, 'CData', greyBackground);  for t = 1:initialPauseFrames  writeVideo(video, getframe(hAx));  end  greyBackgroundCount = 0;  for stage = 1:totalStages  density = densities(stage);  angularSpeed = angularSpeeds(stage);  stageDuration = stageDurations(stage);  numFrames = round(stageDuration \* video.FrameRate);    numCycles = round(density \* 360);  x = linspace(0, 2 \* pi \* numCycles, imageWidth);  sinWave = (sin(x) + 1) / 2;  pattern = uint8(255 \* repmat(sinWave, imageHeight, 1));  shiftAmount = round(imageWidth \* angularSpeed / 360);  originalPattern = pattern;    for t = 1:numFrames  set(hImage, 'CData', pattern);    frame = getframe(hAx);  writeVideo(video, frame);  pattern = circshift(pattern, [0, shiftAmount]);  end  set(hImage, 'CData', greyBackground);  greyBackgroundCount = greyBackgroundCount + 1;  if greyBackgroundCount >= 2  if mod(greyBackgroundCount, 2) == 0  txt = sprintf('S%d(F)', (greyBackgroundCount + 2) / 2);  else  txt = sprintf('S%d(R)', (greyBackgroundCount + 1) / 2);  end  hText = text(hAx, imageWidth/2, imageHeight/2, txt,...  'FontSize', 80, 'Color', [1 1 1],...  'FontWeight', 'bold',...  'HorizontalAlignment', 'center',...  'VerticalAlignment', 'middle');  frame = getframe(hAx);  end  pauseFrames = round(pauseDuration \* video.FrameRate);  for t = 1:pauseFrames  if greyBackgroundCount >= 2  writeVideo(video, frame);  else  writeVideo(video, getframe(hAx));  end  end  if greyBackgroundCount >= 2  end  pattern = originalPattern;  for t = 1:numFrames  set(hImage, 'CData', pattern);  frame = getframe(hAx);  writeVideo(video, frame);  pattern = circshift(pattern, [0, -shiftAmount]);  end  set(hImage, 'CData', greyBackground);  greyBackgroundCount = greyBackgroundCount + 1;  if greyBackgroundCount >= 2  if mod(greyBackgroundCount, 2) == 0  txt = sprintf('S%d(F)', (greyBackgroundCount + 2) / 2);  else  txt = sprintf('S%d(R)', (greyBackgroundCount + 1) / 2);  end  hText = text(hAx, imageWidth/2, imageHeight/2, txt,...  'FontSize', 80, 'Color', [1 1 1],...  'FontWeight', 'bold',...  'HorizontalAlignment', 'center',...  'VerticalAlignment', 'middle');  frame = getframe(hAx);  end  pauseFrames = round(pauseDuration \* video.FrameRate);  for t = 1:pauseFrames  if greyBackgroundCount >= 2  writeVideo(video, frame);  else  writeVideo(video, getframe(hAx));  end  end  if greyBackgroundCount >= 2  delete(hText);  end  fprintf('Stage %d 完成\n', stage);  end  close(video);  close all; |