```
function jf45irv4(input)
# jonfields45@gmail.com
# input.wav is the sample to be processed, pickup left, mic right
diary logfile.txt;
diary on;
# IR generation:
ss = (2^17); # ~3 second segment size for analysis
nstep = 1; # segment step ~3 seconds
nstart = 3; # index into .wav in segments, ~6 seconds
nlast = 9; # up to 7 segments checked for suitable SNR & no clip
count = 0; # initialize good segment count
nzcount = 0; # initialize near zero count
accirfftnnz = zeros(ss,1); # initialize no near zero IR accumulator
window = (.42-.5*cos(2*pi*(0:ss-1)/(ss-1))+.08*cos(4*pi*(0:ss-1)/(ss-1)))'; # Blackman
#
for n = nstart:nstep:nlast # process segments
     start = (((n-1) * ss) + 1);
     finish = (n * ss);
     [s,fs] = audioread([input,'.wav'],[start, finish]); # load segment
     smax = max(max(s));
     clip = (smax > 0.999); # check for clipping
     toolow = (smax < 0.178); # more than 15 dB down
     if(clip == 1)
          ['clipping detected']
          end
     if(toolow == 1)
          ['low SNR detected']
          end
     if (clip == 0 && toolow == 0 && count < 4) # per segment IR
          pickup = s(:,1) .* window;
          mic = s(:,2) .* window;
          pupfft = fft(pickup);
          micfft = fft(mic);
          pupfftnnz = pupfft;
          micfftnnz = micfft;
          nearzero = 10^(-65/20)*abs(max(pupfft)); # ~0 is -65dB from peak
          for m = 1:1:ss
               if (abs(pupfft(m)) < nearzero) # Check near 0 div
                    nzcount = nzcount + 1; # Count near 0s (pos & neg freq)
                    pupfftnnz(m) = 1; # erase near zero
                    micfftnnz(m) = 1; # erase near zero
               end
          end
          irfftnnz = micfftnnz ./ pupfftnnz; # segment IR=FFT(mic)/FFT(PUP)
          accirfftnnz = accirfftnnz + irfftnnz; # accumulate segment IRs
          count = count + 1; # increment processed segment count
     end
if (count == 4)
     break;
     end
end
if(count == 0)
    ['Zero Segments Processed due to Clip/Min errors']
    return;
end
#
irnnz = ifft(accirfftnnz/count); # calc average IR over processed segments
ir2048nnz = irnnz(1:2048); # truncate to 2048
#
avgnzcount = nzcount / (count*2); # pos freq avg near zero count per seg
['Processed Segments = ',num2str(count)]
['Average Near Zero per Segment = ',num2str(avgnzcount)]
['Sample Rate = ',num2str(fs)]
```

```
#
filename = ['jf45ir ',input,'.wav'];
audiowrite(filename,ir2048nnz,fs); # write IR
#
irrms=abs((sum(ir2048nnz.*ir2048nnz))^0.5);
['IR RMS = ',num2str(irrms)]
['IR[0] = ',num2str(ir2048nnz(1))]
#
ir2048nnz7525=(ir2048nnz*0.75); # 75/25 IR
ir2048nnz7525(1)=ir2048nnz7525(1)+(irrms*0.25);
filename = ['jf45ir7525 ',input,'.wav'];
audiowrite(filename,ir2048nnz7525,fs);
#
ir2048nnz5050=(ir2048nnz*0.5); # 50/50 IR
ir2048nnz5050(1)=ir2048nnz5050(1)+(irrms*0.5);
filename = ['jf45ir5050 ',input,'.wav'];
audiowrite(filename,ir2048nnz5050,fs);
#
ir2048nnz00100=(ir2048nnz*0); # bypass IR
ir2048nnz00100(1)=ir2048nnz00100(1)+(irrms*1);
filename = ['jf45irBypass ',input,'.wav'];
audiowrite(filename,ir2048nnz00100,fs);
#
# IR Frequency Plot
fftir2048nnz = fft(ir2048nnz); # FFT IR for frequency plot
y = 20*log(abs(fftir2048nnz(2:1024))); # print frequency plot
x = fs*(1:1023)/2048;
semilogx(x,y);
xlim([20 15000]);
xlabel('Hz');
filename = ['FFTjf45ir ',input,'.jpg'];
ylabel([filename,' dB']);
print(filename,'-djpeg');
#
close;
diary off;
```