

```

199
200 % Split signal into channels with or without downsamp
201 % channels.
202 iChannel = length(hpChannels);
203 if(sampl)
204     while iChannel > 0
205         hpChannelX = imfilter(imfilter(hpChannels{iCh
206             lx', 'replicate', 'conv'), kx, 'replicate
207
208         hpChannelY = imfilter(imfilter(hpChannels{iCh
209             kx', 'replicate', 'conv'),lx, 'replicate'
210
211         lpChannel = interp2(lpChannel,'spline');
212         if mod(size(hpChannelX,1),2) == 0
213             lpChannel = [lpChannel;lpChannel(end,:)];
214         end
215         if mod(size(hpChannelX,2),2) == 0
216             lpChannel = [lpChannel lpChannel(:,end)];
217         end
218
219         lpChannel = imfilter(imfilter(lpChannel,hx','
220             'replicate');
221         lpChannel = hpChannelX + hpChannelY +lpChanne
222         iChannel = iChannel-1;
223     end
224 else
225     while iChannel > 0
226         hpChannelX = imfilter(imfilter(hpChannels{iCh
227             zeroPad(lx,iChannel)', 'replicate', 'conv
228             zeroPad(kx,iChannel), 'replicate', 'conv'
229
230         hpChannelY = imfilter(imfilter(hpChannels{iCh
231             zeroPad(kx,iChannel)', 'replicate', 'conv
232             zeroPad(lx,iChannel), 'replicate', 'conv'
233
234         lpChannel = imfilter(imfilter(lpChannel,zeroP
235             'replicate'),zeroPad(hx,iChannel),'repl
236
237         lpChannel = hpChannelX + hpChannelY +lpChanne
238         iChannel = iChannel-1;
239     end
240 end
241 img = lpChannel;
242 % -----
243
244 % ----- SOFTTHRESH -----
245 function U = softThresh( V, tMax, tMin, sigma, alpha,
246 %SOFTTHRESH performs soft thresholding of the values
247 %   Y = softThresh( X, TMAX, TMIN, S, A, L ) performs
248 %   thresholding of X with the threshold t computed a
249 %
250 %       ( (TMAX-A*(L-1))*S if TMAX-A*(L-1)>TMIN
251 %   t = <
252 %       ( TMIN*S           otherwise
253
254 % Calculate a threshold
255 c = tMax - alpha*(lev-1);

```