Video Magnification

Niels Hoogerwerf (4105761), Vera Hoveling (4591941)

June 2020

1 Getting comfortable: Eulerian video magnification for revealing subtle changes in the world.

1.1 Analyze what is going on spatially in [6].

Each frame consists of three channels. A full Laplacian pyramid is build for each channel of each frame. A separable binomial filter of size five is used to construct the pyramids. The smallest pyramid level possible thus represents a 5x5 image. Essentially, by doing this, the video is downsampled and lowpass-filtered for each frame to reduce both quantization and noise. Then each of the levels of the pyramid is used to boost the subtle signals. This is the first step in processing the video. An example of the first few levels of a Laplacian pyramid can be inspected in Figure 5.

We found that there are three pyramids used: one for each color channel of a frame. These pyramids represent the luminance (one pyramid) and the chrominance (two pyramids) respectively. These color channels map to the YIQ color space (NTSC) that is used, where Y represents the luminance and I and Q are chrominance axes (orange-blue and purple-green respectively).

While the difference between the use of the two pyramids for the two chrominance channels might be less obvious, they just respond to a different range of colors. However, one could see that the difference between the luminance and chrominance pyramid can be important: some edges might be undetectable in just one of the channels. A good example of unresponsiveness of channels can be seen in figures 3 and 4 found in the Appendix.

Each frame is downsampled by a factor of two, to create the smaller pyramid layers subsequently. All pyramids are the maximum possible height. Due to the binomial filter of size five, the minimum height and width of the picture must be at least five pixels, after downsampling to the highest level of the pyramid. The spatial decomposition created by the Laplacian pyramid represents the coarseness of features recognized in each layer. The higher we go into the pyramid, the finer details are omitted and only the more clearly defined edges (with respect to the chrominance/luminance difference) are found. While the paper only mentions the use of Laplacian pyramids to magnify motions and colors, we discovered during implementation that the authors chose a different pyramid for the color magnification. This is discussed further in subsection 1.4.

1.2 Analyze whats going on temporally in [6].

On each spatial band, temporal processing is performed. The time series are considered to be corresponding to the changing value of a pixel, resulting in a frequency for this pixel over time. A bandpass filter is applied to extract the frequency bands of interest. The temporal processing is uniform for all spatial levels: the different frequencies of the intensity of each of the pixels are measured. The end-user can specify which upper and lower bounds are used for the frequencies. In the simplest form one could use two standard first order IIR filters to create the bandpass filter to filter frequencies which are not interesting to amplify. Then, the extracted bandpassed signal is multiplied by a magnification factor α, which is specified by the user. Finally, the magnified signal is added to the original.

This amplification works because of the relation between temporal and spatial changes in a translation in an image: the temporal change in brightness is related to the amount of translation through the gradient of the image, so that if we increase the temporal change in brightness at each point while keeping the spatial gradients, we perceive a larger translation. This is a linear approximation of the true magnified motion. An illustration of this can be found in Figure 1.

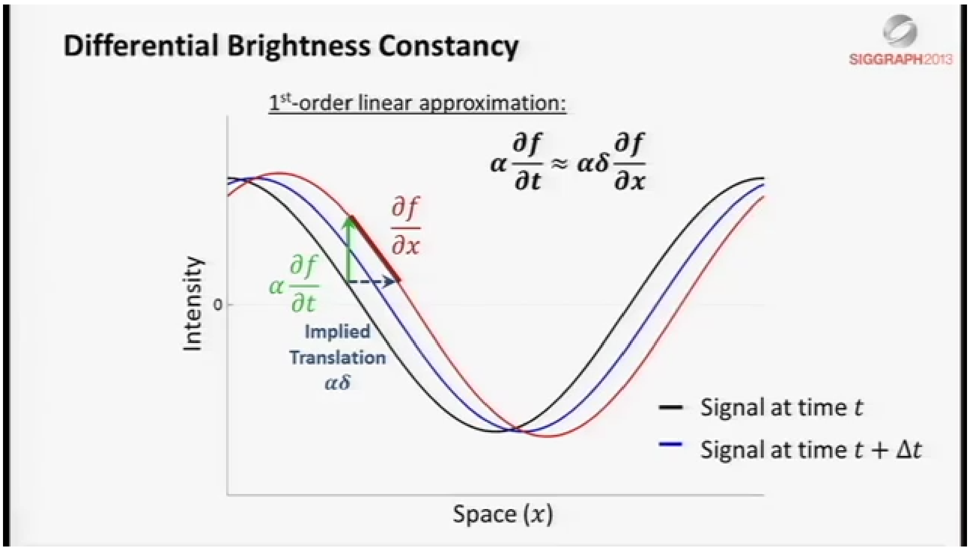


Figure 1: Temporal changes in brightness relate to the translation. Image from [4].

1.3 Briefly describe the difference between Eulerian and Lagrangian motion.

Both the Eulerian and Lagrangian method are often used in fluid flow estimation. Lagrangian methods focus on individual pixels and track their individual trajectory (or particles, in fluid dynamics). In contrast, Eulerian methods focus on the overall motion (i.e. clusters of pixels, objects moving etc.). Lagrangian methods were used for video magnification by [1], before the introduction of [6].

The Lagrangian approach to motion magnification relies on the estimation of motion. This is not just computationally expensive, but for the motion estimation additional techniques are also required (such as image in-painting and motion segmentation) to obtain a reasonable result. These complex steps are omitted in the Eulerian approach, where the variation of pixel values over time is studied (instead of the motion of the pixels).

The Lagrangian approach, however, is more suitable for amplifying large motions than the Eulerian method. This is mainly because the Lagrangian approach tracks motion and thus can predict where pixels should go to. The Eulerian method just simulates linear translation of the pixels and thus the larger the magnified motion the more likely it is that the Eulearian approach will be erroneous.

1.4 Reproduce the results of [6].

The code written for our reproduction can be found in this repository: `https://github.com/asylunatic/MotionMagnification`

All video results can be found in this folder: `https://drive.google.com/drive/folders/16TkfXBRLF8dUByYNN-ds7` `usp=sharing`

A few notes on the implementation that we made: Firstly, our implementation follows the paper [6] in the way the motion is magnified. The Laplacian pyramid is created, the temporal changes are bandpassed and then magnified. However where [6] use an additional formula to change the α for different λ_c, where both are input variables, we chose to just magnify by α.

In addition, after investigating the code provided by [6], we found that color amplification is not achieved by magnifying the Laplacian pyramid. The method used for magnifying color is not discussed in the paper. To still be able to reproduce the results given, we dove into the code. From the given code it became clear that the method used to magnify colors is using the following steps:

1. Create a Gaussian pyramid for each of the frames
2. Bandpass a selected level of the Gaussian pyramid to amplify only color frequencies of interest.
3. Use the amplified color frequencies and add those to the original frame to get a color magnified result.

We decided to follow these steps in our own implementation of the color magnification method as well.

	Run 1	Run 2	Run 3	Run 4	Run 5
Motion magnification	39.4	39.7	38.7	39.3	37.7
Motion magnification ours	86.6	86.6	93.8	96.5	96.9
Color magnification	85.2	82.7	83.5	83.1	83.3
Color magnification ours	54.3	55.6	53.4	54.3	54.4

Table 1: Comparison of the speed of our algorithm and the algorithm of [6].

Lastly, a comparison of the speed of our algorithm and the algorithm of [6] is made in Table 1. While the color magnification is faster in our algorithm, the motion magnification is faster in the implementation of [6]. For the comparison the *baby.mp4* file (as provided by [6]) was used.

1.4.1 Result: Color-magnification video

The original video that we recorded for the color-magnification result is *Vera.mp4*. We used an α value of 1.5 to amplify the frequencies between 0.9 and 1.3 Hz. The frequency bounds were chosen to extract a heart rate between 54 en 78 BPM, which would be in the normal resting heart rate range for a healthy adult. The α value was determined empirically to obtain the best result.

These settings gave us the resulting video *Vera_colormagnified.avi*. We can see in the video that the blood flow moves around much like [6]s face video. However, due to some involuntary head movement by Vera, there is some additional motion amplified, which makes the result a bit blurry sometimes.

1.4.2 Result: Motion-magnification video

For our motion magnification we made a recording of our elderly friend Sammy. The original video that we recorded can be found in the folder as *Sammy.mp4*. We used an α value of 10 to amplify the frequencies between 0.2 and 0.8 Hz. The frequencies were selected to be rather low in order to be able to amplify slower motions like breathing. The α value was initially set to correspond with the α value of 10 for [6]s baby video. We found that this value also worked well for Sammys breathing. The result we obtained can be found in *Sammy_magnifiedmotioneulerian.avi*.

We can clearly see Sammys breathing in the video, but also some amplified motion of the ears and eye. At the end there is some magnified nose movement. As a bonus we had a fly making it's amplified way through the video at the end (one that we had barely noticed in the original video!). As expected from this method, there is a fair amount of noise, but we reduced it significantly by not amplifying the first level of the pyramid.

2 Going to the next phase: Phase-based video motion processing.

2.1 Analyze whats going on spatially in [5].

In [5] there is a different image pyramid used than in [6]. [5] uses the complex steerable pyramid, which allows for independent representation of scale and orientation and thus for a linear multi-scale, multi-orientation image decomposition. This however, is not it's most attractive feature, according to [5]: most interestingly, it also provides measurement of local amplitude and phase. It is this property that enables us to process motion. A visualisation of the different decompositions of one frame by the complex steerable pyramid can be inspected in Figure 2.

The use of quadrature phase filters in the complex steerable pyramid allows us to decompose the image into sinusoids. We can then move the phase of these sinusoids to amplify motion. The fact that we can move the phase, instead of amplifying the difference between color values enables us to obtain a much less noisy result than with the approach by [6]. Note that this phase-based approach is still Eulerian, as we still do the magnification per pixel.

The steerable pyramid is more complex than the Laplacian pyramid: it is more computationally expensive and less intuitive to understand. The steerable pyramid also allows for a much less noisy result: we shift the phase, instead of directly modifying pixel intensities. This results in less noisy result because instead of amplifying noise, the noise is only translated. Another benefit of the ability to shift the phase is that it also allows for a larger correct amplification of movement. And another benefit of the

	Run 1	**Run 2**	**Run 3**
Phase magnification	168.6	201.9	188.8
Phase magnification ours	438.4	449.9	453.5

Table 2: Comparison of the speed of our algorithm and the algorithm of [5]

the phase-shift approach is that it makes it possible to attenuate frequencies (by taking the magnification factor α to be in the range [-1, 0)).

Like the Laplacian pyramid, that we discussed in Section 1.1, the complex steerable pyramid can be applied to all three channels of a video (YIQ). However, [5] mainly applies the steerable pyramid to the luminance channel (and not the chrominance channels).

2.2 Implement [5].

The code written for our reproduction can be found in this repository: `https://github.com/asylunatic/MotionMagnification`. Please note that we named our phase-based method *feestbeest*, which is an inside joke for the Dutch.

All video results can be found in this folder: `https://drive.google.com/drive/folders/16TkfXBRLF8dUByYNN-ds7` `usp=sharing`.

For our implementation, we used the image pyramid library *matlabPyrTools* [2]. In this library, functions are provided for creating complex steerable pyramids. The library version used is the almost the same version that was included in the implementation of [5], with a minor improvement. We found that the library had some troubles with reconstructing the images back from the steerable pyramid. This error came from the pyramid height calculation used in the reconstruction function. Because of that we substituted their pyramid height calculation with our own. We added this improved version in our repository.

In terms of speed, a comparison is made between the two implementations. The results can be found in Table 2. These results were obtained using the *baby.mp4* file provided by [5]. Our implementation proves to be a twice as slow as the implementation given by [5]. This might come from the difference in the implementation of the steerable pyramid.

2.2.1 Result: Color-magnification video

To obtain our color magnification, we used *Niels.mp4*. As the main contribution of [5] is very specifically in the filtering of the phases, the improvement on color-magnification is to first attenuate the phases and then color magnify. This allows for the amplification of color, without the amplification of motion [3]. So first we attenuate the phases in Niels' head. For this we use an α value of -0.99, 4 orientations and remove the frequencies in the range of 1 to 1.9 Hz. The intermediate the result we obtained after this can be found in *Niels_attenuated.avi.* Indeed, we see that Niels is very still.

Then we amplify the colors, using an α value of 1.5 (like in 1.4.1) and frequencies between 0.9 and 1.3 Hz, roughly matching expected heart rate. The amplified-color-without-amplified-motion result can be seen in *Niels_attenuated_colormagnified.avi.* We observe that indeed, Niels appears to be almost static, while a pulsating color change shows in his face.

2.2.2 Result: Motion-magnification video

To obtain our result for the phase-based motion-magnification, we used the same video as in 1.4.2, *Sammy.mp4*, since he did so well in the first approach already. We use an α value of 15 (which is larger than the α we used in 1.4.2, as the phase-based allows for a larger magnification). We used 5 orientations and amplified the frequencies in the range of 1 to 1.9 Hz (same range as in 1.4.2).

The result can be observed in *Sammy_feestbeest.avi.* This is the result we were most excited about. Not only do we see the strong amplification of the breathing, but also his involuntary shakey movements (he is really old). The resuls barely contains any noise and the result is much smoother than what we have seen before. The fly is also much less present than in the first result (1.4.2). Do note that some artifacts are visible at the nose movement, which is too strongly amplified. We could probably have prevented that by sticking to α = 10, like in 1.4.2. However, we chose not do to so as we liked the amplification of the other movements a lot.

References

[1] Ce Liu, Antonio Torralba, William T Freeman, Frédo Durand, and Edward H Adelson. Motion magnification. *ACM transactions on graphics (TOG)*, 24(3):519–526, 2005.

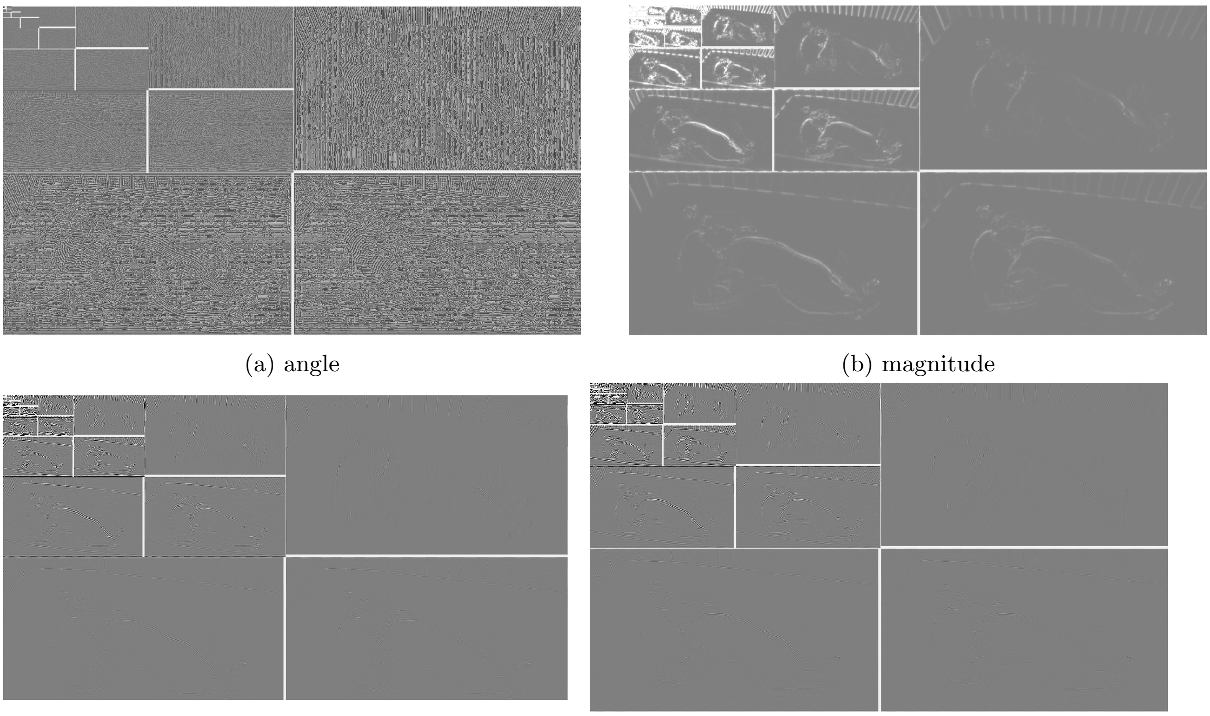
[2] Eero P. Simoncelli. matlabpyrtools. `https://github.com/LabForComputationalVision/matlabPyrTools`, 2009.

[3] Neal Wadhwa, Michael Rubinstein, Fredo Durand, and William T. Freeman. Phase-based video motion processing - supplemental video.

[4] Neal Wadhwa, Michael Rubinstein, Fredo Durand, and William T. Freeman. Phase-based video motion processing (siggraph 2013 presentation).

[5] Neal Wadhwa, Michael Rubinstein, Fredo Durand, and William T. Freeman. Phase-based video motion processing. *ACM Trans. Graph.*, 32(4), July 2013.

[6] Hao-Yu Wu, Michael Rubinstein, Eugene Shih, John Guttag, Frédo Durand, and William Freeman. Eulerian video magnification for revealing subtle changes in the world. *ACM transactions on graphics (TOG)*, 31(4):1–8, 2012.



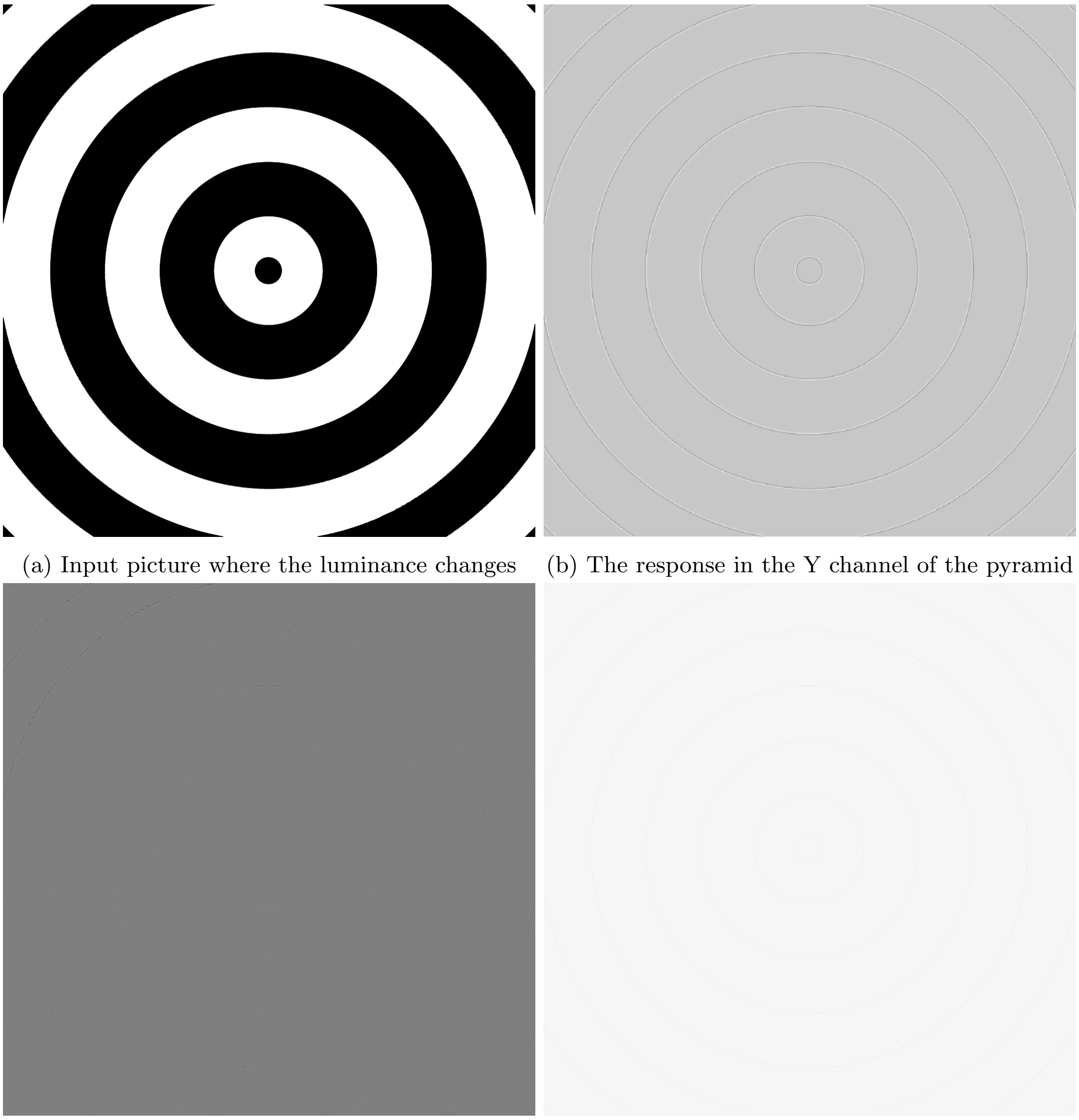
(a) angle

(b) magnitude

(c) real

(d) imaginary

Figure 2: Visualisation of the steerable pyramid as generated for the first frame of the baby video: angle, magnitude, real and imaginary parts



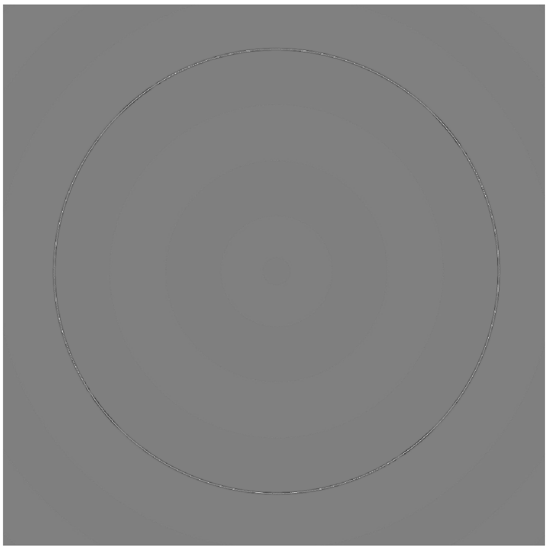
(a) Input picture where the luminance changes

(b) The response in the Y channel of the pyramid

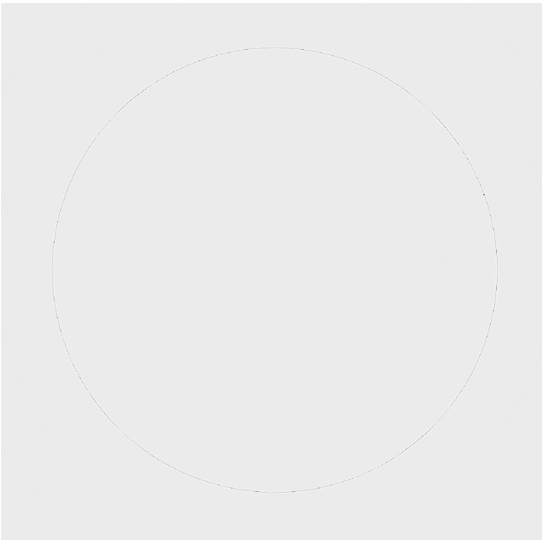
(c) The response in the I channel of the pyramid

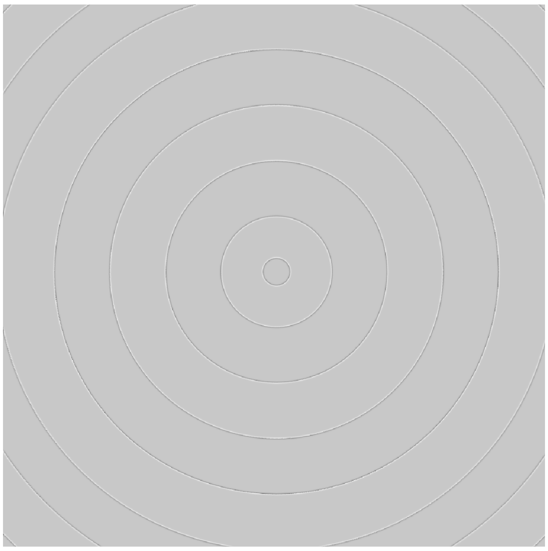
(d) The response in the Q channel of the pyramid

Figure 3: This figure shows that some inputs give responses in different channels. The given input 3a, clearly gives a response in the expected Y channel 3b but it gives no response in the other channels 3c and 3d



(a) Input picture where the chrominance changes (b) The response in the Y channel of the pyramid





(c) The response in the I channel of the pyramid (d) The response in the Q channel of the pyramid

Figure 4: This figure shows that some inputs give responses in different channels. The given input 4a, clearly gives a response in the expected Q channel 4c but it gives almost no response in the other channels 4b and 4d

(a) layer one of the Laplacian pyramid

(b) layer two of the Laplacian pyramid

(c) layer three of the Laplacian pyramid

(d) layer four of the Laplacian pyramid

Figure 5: Overview of the bottom four layers of the Laplacian pyramid for the luminance, one can see that each higher located layer only shows the courser edges