Space-Variant Spatio-Temporal Filtering of Video for Gaze Visualization and Perceptual Learning

Michael Dorr*
Institute for Neuro- and Bioinformatics
University of Lübeck

Halszka Jarodzka†
Knowledge Media Research Center
Tübingen

Erhardt Barth‡
Institute for Neuro- and Bioinformatics
University of Lübeck

Abstract

We introduce an algorithm for space-variant filtering of video based on a spatio-temporal Laplacian pyramid and use this algorithm to render videos in order to visualize prerecorded eye movements. Spatio-temporal contrast and colour saturation are reduced as a function of distance to the nearest gaze point of regard, i.e. non-fixated, distracting regions are filtered out, whereas fixated image regions remain unchanged. Results of an experiment in which the eye movements of an expert on instructional videos are visualized with this algorithm, so that the gaze of novices is guided to relevant image locations, show that this visualization technique facilitates the novices' perceptual learning.

CR Categories: I.4.9 [Image Processing and Computer Vision]: Applications; I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering; I.4.10 [Image Processing and Computer Vision]: Image Representation—Multidimensional; K.3.1 [Computers and Education]: Computer Uses in Education—Computer-assisted instruction (CAI)

Keywords: gaze visualization, space-variant filtering, spatio-temporal Laplacian pyramid, perceptual learning

1 Introduction

Humans move their eyes around several times per second to successively sample visual scenes with the high-resolution centre of the retina. The direction of gaze is tightly linked to attention, and what people perceive ultimately depends on where they look [Stone et al. 2003]. Naturally, the ability to record eye movement data led to the need for meaningful visualizations. One-dimensional plots of the horizontal and vertical components of eye position over time have been in use since the very first gaze recording experiments (Delabarre [1898] affixed a small cap on the cornea to transduce eye movements onto a rotating drum, using plaster of Paris as glue). Such plots are useful for detailed quantitative analyses, but not very intuitively interpreted. Other tools supporting interpretation of the data include the visualization of gaze density by means of clustered gaze samples [Heminghous and Duchowski 2006] or the visualization of other features such as fixation duration [Ramloll et al. 2004].

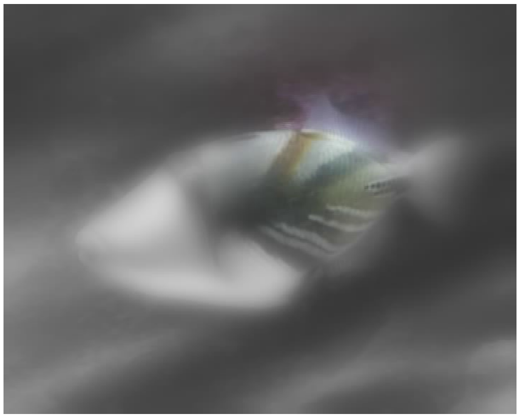

Figure 1: *Stillshot from an instructional video on classifying fish locomotion patterns. The eye movements of the expert giving voice-over explanations are visualized by space-variant filtering on a spatio-temporal Laplacian pyramid: spatio-temporal contrast and colour saturation are reduced in unattended areas. This visualization technique aids novices in acquiring the expert's perceptual skills.*

Better suited for visual inspection are approaches that use the stimulus and enrich it with eye movement data; in the classical paper of Yarbus [1967], gaze traces overlaid on the original images immediately show the regions that were preferentially looked at by the subjects. Because of the noisy nature of both eye movements and their measurements, there is also an indirect indication of fixation duration (traces are denser in areas of longer fixation). However, such abstract information can also be extracted from the raw data and presented in condensed form: for example, bars of different size are placed in a three-dimensional view of the original stimulus to denote fixation duration in [Lankford 2000]; in a more application-specific manner, Špakov and Räihä [2008] annotate text with abstract information on gaze behaviour for the analysis of translation processes.

Another common method is the use of so-called fixation maps [Velichkovsky et al. 1996; Wooding 2002]. Here, a probability density map is computed by the superposition of Gaussians, each centred at a single fixation (or raw gaze sample), with a subsequent normalization step. Areas that were fixated more often are thus assigned higher probabilities; by varying the width of the underlying Gaussians, it is possible to vary the distance up to which two fixations are considered similar. Based on this probability map, the stimulus images are processed so that for example luminance is gradually reduced in areas that received little attention; so-called heat maps mark regions of interest with transparently overlaid colours. In [Špakov and Miniotas 2007], the authors add "fog" to render visible only the attended parts of the stimulus.

*e-mail: dorr@inb.uni-luebeck.de
†e-mail: h.jarodzka@iwm-kwmrc.de
‡e-mail: barth@inb.uni-luebeck.de

For dynamic stimuli, such as movies, all the above techniques can be applied as well; one straightforward extension from images to image sequences would be to apply the fixation map technique to every video frame individually. Care has to be taken, however, to appropriately filter the gaze input in order to ensure a smooth transition between video frames.

In this paper, we present an algorithm to visualize dynamic gaze density maps by locally modifying spatio-temporal contrast on a spatio-temporal Laplacian pyramid. In regions of low interest, spectral energy is reduced, i.e. edge and motion intensity are dampened, whereas regions of high interest remain as in the original stimulus. Conceptually, this algorithm is related to gaze-contingent displays simulating visual fields based on Gaussian pyramids [Geisler and Perry 2002; Nikolov et al. 2004]; in these approaches, however, only fine spatial details were blurred selectively. In earlier work, we have already extended these algorithms to the temporal domain [Böhme et al. 2006], and we here combine both spatial and temporal filtering. Furthermore, the work presented here goes beyond blurring, i.e. the specification of a single cutoff frequency per output pixel, and allows to assign individual weights to each frequency band. This means that, for example, spatio-temporal contrast can be modified while leaving details intact. However, although our implementation of this algorithm achieves more than 50 frames per second rendering performance on a commodity PC, it cannot be used for real-time gaze-contingent applications, because there all levels of the underlying pyramid need to be upsampled to full temporal resolution for every video frame. Its purpose is the off-line visualization of prerecorded gaze patterns. A computationally much more challenging version of a spatio-temporal Laplacian pyramid that is suitable also for gaze-contingent displays is the topic of a forthcoming paper.

Pyramid-based rendering as a function of gaze has been shown to have a guiding effect on eye movements [Dorr et al. 2008; Barth et al. 2006]. To further demonstrate the usefulness of our algorithm, we will present some results from a validation experiment in which students received instructional videos either with or without a visualization of the eye movements of an expert watching the same stimulus. Results show that the visualization technique presented here indeed facilitates perceptual learning and improves students' later visual search performance on novel stimuli.

2 Laplacian Pyramid in Space and Space-Time

The so-called Laplacian pyramid serves as an efficient bandpass representation of an image [Burt and Adelson 1983]. In the following section, we will briefly review its application to images and then extend the algorithm to the spatio-temporal domain. We will here use an isotropic pyramid, i.e. all spatial and temporal dimensions are treated equally; this results in a bandpass representation in which e.g. low spatial and low temporal and high spatial and high temporal frequencies are represented together, respectively. For a finer-grained decomposition of the image sequence into spatio-temporal frequency bands, an anisotropic Laplacian pyramid could be used instead. Using such a decomposition, one could also obtain frequency bands of high spatial but low temporal frequencies etc. For a straightforward implementation, one might first create a spatial pyramid for each frame of the input sequence, then decompose each level of that spatial pyramid in time (as in Section 2.2.2, but omitting the spatial up- and downsampling). However, the finer spectral resolution comes at the cost of a significantly increased number of pixels that need to be stored and processed; this increase is on the order of $1.16 \cdot T$ times as many pixels for an anisotropic pyramid with T temporal levels.

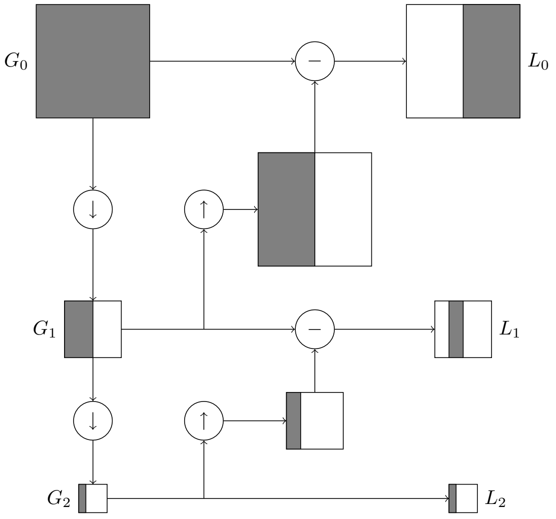

Figure 2: *Analysis phase of a Laplacian pyramid in space. Based on the Gaussian pyramid on the left side, which stores successively smaller image versions (with higher-frequency content successively removed), differences of Gaussian pyramid levels are formed to obtain individual frequency bands (right side). To be able to form these differences, lower levels have to be upsampled before subtraction (middle). The gray bars indicate – relative to the original spectrum – what frequency band is stored in each image. The extension into the temporal domain results in lower frame rates for the smaller video versions (not shown).*

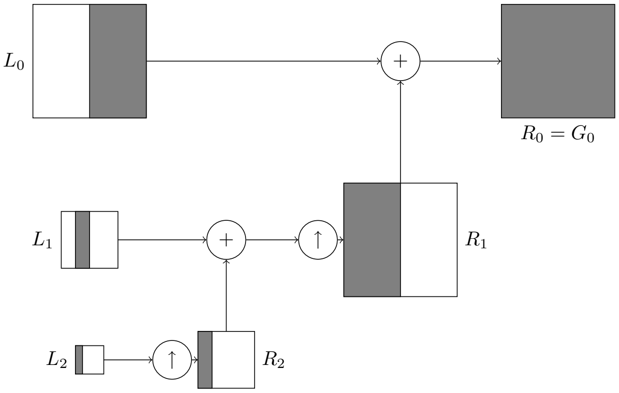

Figure 3: *Synthesis phase of a Laplacian pyramid in space. The Laplacian levels are iteratively upsampled to obtain a series of reconstructed images $R_N, R_{N-1}, \ldots, R_0$ with increasing cutoff frequencies. If the L_n remain unchanged, R_0 is an exact reproduction of the original input image G_0.*

2.1 Spatial Domain

The Laplacian pyramid is based on a Gaussian multiresolution pyramid, which stores successively smaller versions of an image; usually, resolution is reduced by a factor of two in each downsampling step (for two-dimensional images, the number of pixels is thus reduced by a factor of four). Prior to each downsampling step, the image is appropriately lowpass filtered so that high-frequency content is removed; the lower-resolution downsampling result then fulfills the conditions of the Nyquist theorem and can represent the (filtered) image without aliasing. For a schematic overview, we re-

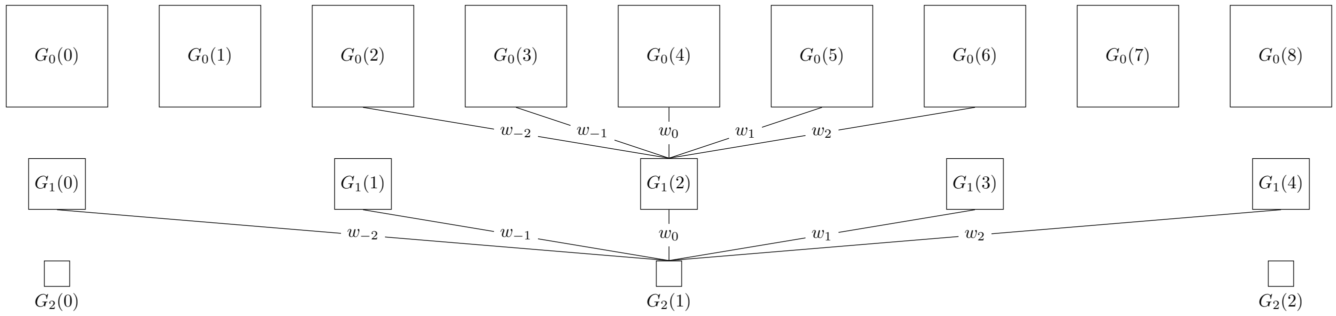

Figure 4: *Spatio-temporal Gaussian pyramid with three levels and $c = 2$. Lower levels have reduced resolution both in space and in time. Note that history and lookahead video frames are required because of the temporal filter with symmetric kernel, e.g. computation of $G_2(1)$ depends on $G_1(4)$, which in turn depends on $G_0(6)$.*

fer to Figures 2 and 3.

2.2 Spatio-Temporal Domain

The Gaussian pyramid algorithm was first applied to the temporal domain by Uz et al. [1991]. In analogy to dropping every other pixel during the downsampling step, every other frame of an image sequence is discarded to obtain lower pyramid levels (cf. Figure 4). In the temporal domain, however, the problem arises that the number of frames to process is not necessarily known in advance and is potentially infinite; it is therefore not feasible to store the whole image sequence in memory. Nevertheless, video frames from the past and the future need to be accessed during the lowpass filtering step prior to the downsampling operation; thus, care has to be taken which frames to buffer. In the following, we will refer to these frames as history and lookahead frames, respectively.

Both for the subtraction of adjacent Gaussian pyramid levels (to create Laplacian levels) and for the reconstruction step (in which the Laplacian levels are recombined), lower levels first have to be upsampled to match the resolution of the higher level. Following these upsampling steps, the results have to be filtered to interpolate at the inserted pixels and frames; again, history and lookahead video frames are required. We will now describe these operations in more detail and analyse the number of video frames to be buffered.

2.2.1 Notation

The sequence of input images is denoted by $I(t)$; input images have a size of W by H pixels and an arbitrary number of colour channels (individual channels are treated separately). A single pixel at location (x, y) and time t is referred to as $I(t)(x, y)$; in the following, operations on whole images, such as addition, are to be applied pixelwise to all pixels.

The individual levels of a Gaussian multiresolution pyramid with $N + 1$ levels are referred to as $G_k(t)$, $0 \leq k \leq N$. The highest level G_0 is the same as the input sequence; because of the spatio-temporal downsampling, lower levels have fewer pixels and a lower frame rate, so that $G_k(n)$ has a spatial resolution of $W/2^k$ by $H/2^k$ pixels and corresponds to the same point in time as $G_0(2^k n)$. Spatial up- and downsampling operations on an image I are denoted as $\uparrow [I]$ and $\downarrow [I]$, respectively. For time steps t that are not a multiple of 2^N, not all pyramid levels have a corresponding image $G_k(t/2^k)$. Therefore, C_t denotes the highest index of levels with valid images at time t, i.e. C_t is the largest integer with $C_t \leq N$ and $t \bmod 2^{C_t} = 0$. Similar to the Gaussian levels G_k, we refer to the levels of the Laplacian pyramid as $L_k(t)$, $0 \leq k \leq N$ (again, resolution is reduced by a factor of two in all dimensions with increasing k); the intermediate steps during the iterative reconstruction of the original signal are denoted as $R_k(t)$.

The temporal filtering which is required for temporal down- and upsampling introduces a latency (see next sections). The number of lookahead items required on level k is denoted by λ_k for the analysis phase and by Λ_k for the synthesis phase.

2.2.2 Analysis Phase

To compute the Laplacian levels, the Gaussian pyramid has to be created first (see Figure 2). The relationship of different Gaussian levels is shown in Figure 4; lower levels are obtained by lowpass filtering and spatially downsampling higher levels:

$$G_{k+1}(n) = \sum_{i=-c}^{c} w_i \cdot \downarrow [G_k(2n - i)] \Bigg/ \sum_{i=-c}^{c} w_i \;.$$

We here use a binomial filter kernel $(1, 4, 6, 4, 1)$ with $c = 2$.

The Laplacian levels are then computed as differences of adjacent Gaussian levels (the lowest level L_N is the same as the lowest Gaussian level G_N); before performing the subtraction, the lower level has to be brought back to a matching resolution again by inserting zeros (blank frames) to upsample and subsequent lowpass filtering. In practice, the inserted frames can be ignored and their corresponding filter coefficients are set to zero:

$$L_k(n) = G_k(n) - \uparrow \left[\sum_{i \in P(n)} w_i \cdot G_{k+1}\left(\frac{n-i}{2}\right) \Bigg/ \sum_{i \in P(n)} w_i \right] ,$$

with $P(n) = \{j = -c, \ldots, c \mid (n - j) \bmod 2 = 0\}$ giving the set of valid images on the lower level.

Based on these equations, we can now derive the number of lookahead items required for the generation of the Laplacian. For the upsampling of lower Gaussian levels, we need a lookahead of $\beta = \lfloor \frac{c+1}{2} \rfloor$ images on each level, with $\lfloor\rfloor$ denoting floating-point truncation. Starting on the lowest level G_N, this implies that $2\beta + c$ images must be available on level G_{N-1} during the downsampling phase; we can repeatedly follow this argument and obtain $\lambda_k = 2^{N-k} \cdot (\beta + c) - c$ as the number of required lookahead images for level k.

2.2.3 Synthesis Phase

Turning now to the synthesis phase of the Laplacian pyramid, we note from Figure 3 that the Laplacian levels are sucessively upsam-

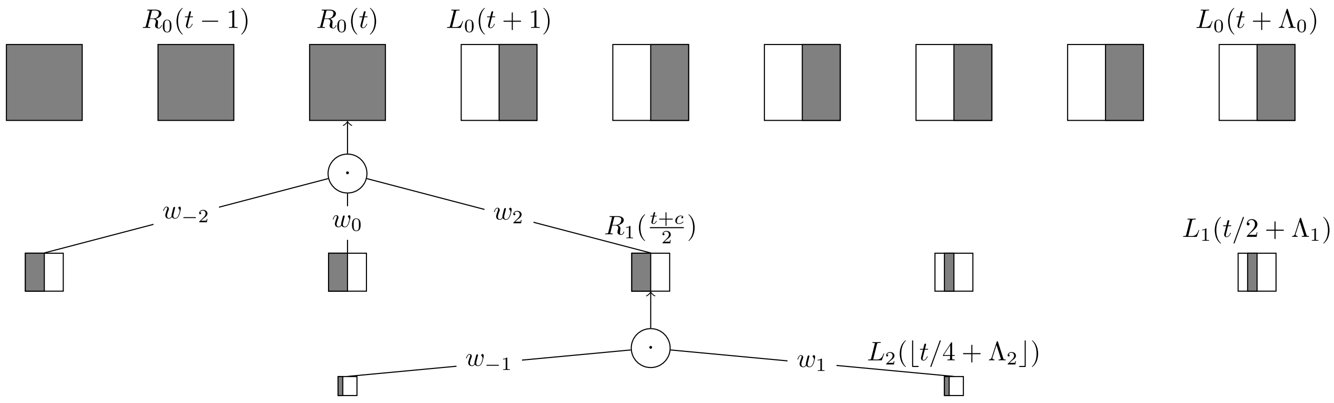

Figure 5: *Synthesis step of spatio-temporal Laplacian pyramid. Here shown are both the Laplacian levels L_i and the (partially) reconstructed images R_i, which are based on lower levels with indices $\geq i$; in practice, the same buffers can be used for both R and L. For example, to compute the reconstruction $R_0(t)$ of the original image, we have to add $L_0(t)$ to a spatio-temporally upsampled version of R_1. The second level L_1 is combined with the upsampling result of L_2 in $R_1(\frac{t+c}{2}) = R_1(\frac{t}{2}+\beta_1)$ (see pseudocode). In this schematic overview, new frames are added on the right side and shifted leftwards with time.*

pled and added up to reconstruct the original image; this simply is the inverse of the "upsample–and–subtract" operation during the analysis phase. On the lowest level, $R_N(n) = L_N(n)$; for higher levels, the intermediate reconstructed images are computed as

$$R_k(n) = L_k(n) + \sum_{i\in P(n)} w_i \Bigg\uparrow \left[R_{k+1}\left(\frac{n-i}{2}\right) \Big/ \sum_{i\in P(n)} w_i \right]. \tag{1}$$

Clearly, a further latency is incurred between the point in time for which bandpass information and the reconstructed or filtered image are available. Similar to the study of the analysis phase in the previous section, we can compute the number Λ_k of required lookahead items on each level by induction. On the lowest level L_N, again $\beta = \lfloor\frac{c+1}{2}\rfloor$ images are required for the upsampling operation, which corresponds to 2β images on level $N-1$. As can be seen in Figure 5, the result of the upsampling operation is added to the β-th lookahead item on level $N-1$, so that $\Lambda_{N-1} = 3\beta$. Repeating this computation, we obtain $\Lambda_k = (2^{N+1-k}-1)\cdot\beta$; for L_0, however, no further upsampling is required, so it is possible to reduce the lookahead on the highest level to $\Lambda_0 = (2^{N+1}-2)\cdot\beta$.

In practice, we do not need to differentiate explicitly between L and R; the same buffers can be used for both L and R images. Care has to be taken then not to perform a desired modification of a given Laplacian level on a buffer that already contains information from lower levels as well (i.e. an R image).

2.3 Pseudocode and Implementation

We are now ready to bring together the above observations and put them into pseudocode, see Algorithms 1 and 2. Based on $P(n)$ above, the index function that determines which images are available on lower levels in the following is $P_k(t) = \{j = -c,\ldots,c \mid \left(\frac{t}{2^k}+\beta_k-j\right) \bmod 2 = 0\}$. In the synthesis phase, the image offset β_k at which the recombination L and R takes place can be set to zero on the highest level; we therefore use $\beta_k = \beta$ for $k > 0$, $\beta_0 = 0$.

From the pseudocode, a buffering scheme for the implementation directly follows. First, images from the Gaussian pyramid have to be stored; each level k needs at least λ_k lookahead images, one current image, and β_k history. Trading memory requirements for computational costs, it is also possible to keep all images of the Gaussian pyramid in memory twice, once in the "correct" size and once in the downsampled version; for each frame of the input video, only one downsampling operation has to be executed then. In analogy to the Gaussian levels, both the Laplacian and the (partially) reconstructed levels L and R can be held together in one buffer per level k with Λ_k lookahead, one current image, and the β_k history.

In practice, the output of the pyramid can be accessed only with a certain latency because of the symmetric temporal filters that require video frames from the future. Input images are fed into lookahead position λ_0 of buffer G_0, and images are shifted towards the "current" position by one position for every new video frame. This means that only λ_0 many time steps after video frame $I(t_0)$ has been added, the Gaussian images G_0 to G_N that represent $I(t_0)$ at various spatio-temporal resolutions are available in the "current" positions of the Gaussian buffers. The resulting differences L_0 to L_N then are stored at the lookahead positions Λ_0 to Λ_N of the Laplacian buffers, respectively; here, different frequency bands can be accessed both for analysis and modification. Only Λ_0 time steps later does the input image I reappear after pyramid synthesis; overall, this leads to a pyramid latency between input and output of $\lambda_0 + \Lambda_0$ time steps.

The necessary buffering and the handling of lookahead frames could be reduced and simplified if causal filters were used; a further possibility to efficiently filter in time without lookahead is to use temporally recursive filters. However, any non-symmetry in the filters will introduce phase shifts. Particularly in the case of space-variant filtering (see below), this would produce image artefacts (such as a pedestrian with disconnected – fast – legs and – relatively slow – upper body).

3 Gaze Visualization

In the previous section, we described the analysis and synthesis phase of a spatio-temporal Laplacian pyramid. However, the result of the synthesis phase is a mere reconstruction of the original image sequence; we want to filter the image sequence based on a list of gaze positions instead.

Algorithm 1 Pseudocode for one time step of the pyramid analysis phase.

```
Input:  $t$                     Time step to update the pyramid for
        $G_0, \ldots, G_N$      Levels of the Gaussian pyramid
        $L_0, \ldots, L_N$      Levels of the Laplacian pyramid

$C_t = \max(\{\gamma \in \mathbb{N} \mid 0 \leq \gamma \leq N, t \bmod 2^\gamma = 0\})$
                                                ▷ Gaussian pyramid creation
$G_0(t + \lambda_0) = I(t + \lambda_0)$
for $k = 1, \ldots, C_t$ do
    $G_k\left(\frac{t}{2^k} + \lambda_k\right) = \sum_{i=-c}^{c} w_i \cdot \downarrow\left[G_{k-1}\left(\frac{t}{2^{k-1}} + 2\lambda_k - i\right)\right] \Big/ \sum_{i=-c}^{c} w_i$
end for
                                                ▷ Laplacian pyramid creation
for $k = 0, \ldots, C_t$ do
    if $k = N$ then
        $L_N\left(\frac{t}{2^N} + \Lambda_N\right) = G_N\left(\frac{t}{2^N} + \Lambda_N\right)$
    else
        $L_k\left(\frac{t}{2^k} + \Lambda_k\right) = G_k\left(\frac{t}{2^k} + \Lambda_k\right) - \uparrow\left[\sum_{i \in P_k(t)} w_i \cdot G_{k+1}\left(\frac{t}{2^{k+1}} + \frac{\Lambda_k - i}{2}\right)\right] \Big/ \sum_{i \in P_k(t)} w_i$
    end if
end for
```

3.1 Space-Variant Pyramid Synthesis

We now introduce the concept of a *coefficient map* that indicates how spectral energy should be modified in each frequency band at each pixel of the output image sequence. We denote the coefficient map for level k at time t with $W_k(t)$; the W_k have the same spatial resolution as the corresponding L_k, i.e. $W/2^k$ by $H/2^k$ pixels.

To bandpass-filter the image sequence, the Laplacian levels L_k are simply multiplied pixel-wise with the W_k prior to the recombination into R_k.

Based on the pseudocode (Algorithm 3), we can see that coefficient maps for different points in time are applied to the different levels in each synthesis step of the pyramid; this follows from the iterative recombination of L into the reconstructed levels. In practice, a more straightforward solution is to apply coefficient maps corresponding to one time t to the farthest lookahead item Λ_k of each level L_k (i.e. right after subtraction of adjacent Gaussian levels).

As noted before, in the following validation experiment we will use the same coefficient map for all levels (for computational efficiency, however, coefficient maps for lower levels can be stored with fewer pixels). In principle, this means that a similar effect could be achieved by computing the mean pixel intensity of the whole image sequence and then, depending on gaze position, smoothly blending between this mean value and each video pixel. However, for practical reasons, the lowest level of the pyramid does not represent the "true" DC (the mean of the image sequence), but merely a very strongly lowpass-filtered video version; this means that some coarse spatio-temporal structure remains even in regions where all contrast in higher levels is removed by setting the coefficient map to zero. The temporal multiresolution character of the pyramid also adds smoothness to changes in the coefficient maps over time; because temporal levels are updated at varying rates, such changes are introduced gradually. Finally, by using different coefficient maps for each level, it is trivially possible to highlight certain frequency bands, which is impossible based on a computation of the mean alone.

Algorithm 2 Pseudocode for one time step of the pyramid synthesis phase.

```
Input:  $t$                     Time step to update the pyramid for
        $G_0, \ldots, G_N$      Levels of the Gaussian pyramid
        $L_0, \ldots, L_N$      Levels of the Laplacian pyramid

$C_t = \max(\{\gamma \in \mathbb{N} \mid 0 \leq \gamma \leq N, t \bmod 2^\gamma = 0\})$
for $k = C_t, \ldots, 0$ do
    if $k = N$ then
        $R_N\left(\frac{t}{2^N} + \beta_N\right) = L_N\left(\frac{t}{2^N} + \beta_N\right)$
    else
        $R_k\left(\frac{t}{2^k} + \beta_k\right) = L_k\left(\frac{t}{2^k} + \beta_k\right) + \uparrow\left[\sum_{i \in P_k(t)} w_i \cdot R_{k+1}\left(\frac{t}{2^{k+1}} + \frac{\beta_k - i}{2}\right)\right]$
    end if
end for
```

4 Attentional Guidance

Pyramid-based rendering of video as a function of gaze has been shown to have a guiding effect on eye movements. For example, the introduction of peripheral temporal blur on a gaze-contingent display reduces the number of large-amplitude saccades [Barth et al. 2006], even though the visibility of such blur is low [Dorr et al. 2005]. Using a real-time gaze-contingent version of a spatial Laplacian pyramid, locally reducing (spatial) spectral energy at likely fixation points also changes eye movement characteristics [Dorr et al. 2008].

In the following, we will therefore briefly summarize how our gaze visualization algorithm can be applied in a learning task to guide the student's gaze. For further details of this experiment, we refer to [Jarodzka et al. 2010b].

4.1 Perceptual Learning

In many problem domains, experts develop efficient eye movement strategies because the underlying problem requires substantial visual search. Examples include the analysis of radiograms [Lesgold et al. 1988], driving [Underwood et al. 2003], and the classification of fish locomotion [Jarodzka et al. 2010a]. In order to aid novices in acquiring the efficient eye movement strategies of an expert, it is possible to use cueing to guide their attention towards relevant stimulus locations; however, it often remains unclear where and how to cue the user. Van Gog et al. [2009] guided attention during problem-solving tasks by directly displaying the eye move-

Algorithm 3 Pseudocode for one time step of the space-variant synthesis phase.

```
Input:  $t$                      Time step to update the pyramid for
        $G_0, \ldots, G_N$       Levels of the Gaussian pyramid
        $L_0, \ldots, L_N$       Levels of the Laplacian pyramid
        $W_0, \ldots, W_N$       Coefficient maps

  $C_t = \max(\{\gamma \in \mathbb{N} \mid 0 \leq \gamma \leq N, t \bmod 2^\gamma = 0\})$
  for $k = C_t, \ldots, 0$ do
      if $k = N$ then
          $$R_N\left(\frac{t}{2^N} + \beta_N\right) = L_N\left(\frac{t}{2^N} + \beta_N\right)$$
      else
          $$R_k\left(\frac{t}{2^k} + \beta_k\right) = W_k\left(x, y, \frac{t}{2^k} + \beta_k\right) \cdot L_k\left(x, y, \frac{t}{2^k} + \beta_k\right) + \Bigg\uparrow \left[\sum_{i \in P_k(t)} w_i \cdot R_{k+1}\left(\frac{t}{2^{k+1}} + \frac{\beta_k - i}{2}\right)\right]$$
      end if
  end for
```

ments of an expert made during performing the same task on modeling examples, but found that the attentional guidance actually decreased novices' subsequent test performance instead of facilitating the learning process. One possible explanation of this effect could be that the chosen method of guidance (a red dot at the experts' gaze position that grew in size with fixation duration) was not optimal because the gaze marker covered exactly those visual features it was supposed to highlight, and its dynamical nature might have distracted the observers. To avoid this problem, we here use the space-variant filtering algorithm presented in the previous sections to render instructional videos such that the viewer's attention is guided to those areas that were attended by the expert. However, instead of altering these attended areas, we decrease spatio-temporal contrast (i.e. edge and motion intensity) elsewhere, in order to increase the relative visual saliency of the problem-relevant areas without covering them or introducing artefacts.

4.2 Stimulus Material and Experimental Setup

Eight videos of different fish species with a duration of 4 s each were recorded, depicting different locomotion patterns. They had a spatial resolution of 720 by 576 pixels and a frame rate of 25 frames per second. Four of these videos were shown in a continuous loop to an expert on fish locomotion (a professor of marine zoology) and his eye movements were collected using a Tobii 1750 remote eye tracker running at 50 Hz. Simultaneously, a spoken didactical explanation of the locomotion pattern (i.e. how different body parts moved) was recorded. These four videos were shown to 72 subjects (university students without prior task experience) in a training phase either as-is, with the expert's eye movements marked by a simple yellow disk at gaze position, or with attentional guidance by the pyramid-based contrast reduction. In the subsequent test or recall phase, the remaining four videos were shown to the subjects without any modification. After presentation, subjects had to apply the knowledge acquired during the training phase and had to name and describe the locomotion pattern displayed in each test video; the number of correct answers yielded a performance score.

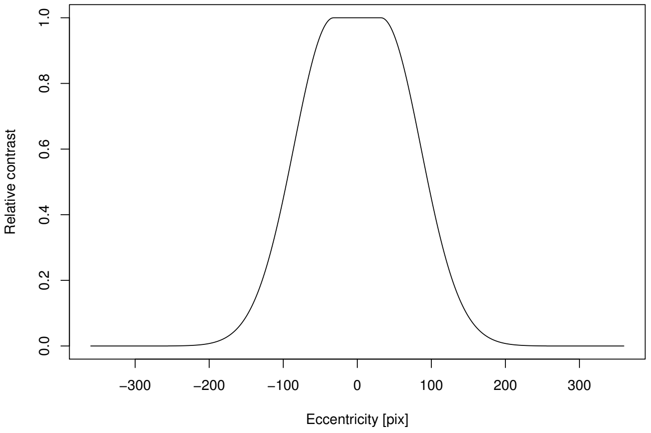

Figure 6: *Eccentricity-dependent resolution map: at centre of fixation, spectral energy remains the same; energy is increasingly reduced with increasing distance from gaze position.*

4.3 Gaze Filtering

Functionally, a sequence of eye movements consists of a series of fixations, where eye position remains constant, and saccades, during which eye position changes rapidly (smooth pursuit movements here can be understood as fixations where gaze position remains constant on a moving object). In practice, however, the eye position as measured by the eye tracker hardly ever stays constant from one sample to the next; the fixational instability of the oculomotor system, minor head movements, and noise in the camera system of the eye tracker all contribute to the effect that the measured eye position exhibits a substantial jitter. If this jitter were to be replayed to the novice, such constant erratic motion might distract the observer from the very scene that gaze guidance is supposed to highlight. In order to reduce the jitter, raw gaze data was filtered with a temporal Gaussian lowpass filter with a support of 200 ms and a standard deviation of 42 ms.

4.4 Space-Variant Filtering and Colour Removal

A Laplacian pyramid with five levels was used; coefficient maps were created in such a way that the original image sequence was reconstructed faithfully in the fixated area (the weight of all levels during pyramid synthesis was set to 1.0) and spatio-temporal changes were diminished (all level weights set to 0.0) in those areas that the expert had only seen peripherally. On the highest level, the first zone was defined by a radius of 32 pixels around gaze position and weights were set to 0.0 outside a radius of 256 pixels; these radii approximately corresponded to 1.15 and 9.2 degrees of visual angle, respectively. In parafoveal vision, weights were gradually decreased from 1.0 to 0.0 for a smooth transition, following a Gaussian falloff with a standard deviation of 40 pixels (see Figure 6). Furthermore, these maps were produced not only by placing a mask at the current gaze position in each video frame; instead, masks for all gaze positions of the preceding and following 300 ms were superimposed and the coefficient map was then normalized to a maximum of 1.0. During periods of fixation, this superposition had little or no effect; during saccades, however, this procedure elongated the radially symmetric coefficient map along the direction of the saccade. Thus, the observer was able to follow the expert's saccades and unpredictable large displacements of the unmodified area were prevented. Finally, colour saturation was also removed from non-attended areas similar to the reduction of spectral energy; here, complete removal of colour started outside a radius of 384 pixels around gaze, and the Gaussian falloff in the transition area had a standard deviation of 67 pixels. Note that these parameters were determined rather informally to find a reasonable trade-off between a focus that would be too restricted (if the focus were only a few pixels wide, discrimination of relevant features would be impossible) and a wide focus that would be without effect (if the unmodified area encompassed the whole stimulus). As such, these parameters are likely to be specific to the stimulus material used here. For a thorough investigation of the visibility of peripherally removed colour saturation using a gaze-contingent display, we refer to Duchowski et al. [2009].

For an example frame, see Figure 1; a demo video is available online at http://www.gazecom.eu/demo-material.

4.5 Results

Previous research has already shown that providing a gaze marker in the highly perceptual task of classifying fish locomotion facilitates perceptual learning: subjects look at relevant movie regions for a longer time and take less time to find relevant locations after stimulus onset, which in turn results in higher performance scores in subsequent tests on novel stimuli [Jarodzka et al. 2009]. The gaze visualization technique presented here does not cover these relevant locations; subjects' visual search performance is improved even beyond that obtained with the simple gaze marker. Time needed to find relevant locations after stimulus onset decreases by 21.26% compared to the gaze marker condition and by 27.53% compared to the condition without any guidance. Moreover, dwell time on the relevant locations increases by 7.25% compared to the gaze marker condition and by 48.82% compared to the condition without any guidance. For a more in-depth analysis see [Jarodzka et al. 2010b].

5 Conclusion

We have presented a novel algorithm to perform space-variant filtering of a movie based on a spatio-temporal Laplacian pyramid. One application is the visualization of eye movements on videos; spatio-temporal contrast is modified as a function of gaze density, i.e. spectral energy is reduced in regions of low interest. In a validation experiment, subjects watched instructional videos on fish locomotion either with or without visualization of the eye movements of an expert. We were able to show that on novel test stimuli, subjects who had received such information performed better than subjects who had not benefited from the expert's eye movements during training, and that the gaze visualization technique presented here facilitated learning better than a simple gaze display (yellow gaze marker). In principle, any visualization technique that reduces the relative visibility of those regions not attended by the expert might have a similar effect; our choice for this particular technique was motivated by our work on eye movement prediction [Dorr et al. 2008; Vig et al. 2009], which shows that spectral energy is a good predictor for eye movements. Ultimately, we intend to use similar techniques in a gaze-contingent fashion in order to guide the gaze of an observer [Barth et al. 2006].

Acknowledgements

Our research has received funding from the European Commission within the project GazeCom (contract no. IST-C-033816) of the 6th Framework Programme. All views expressed herein are those of the authors alone; the European Community is not liable for any use made of the information. Halszka Jarodzka was supported by the Leibniz-Gemeinschaft within the project "Resource-adaptive design of visualizations for supporting the comprehension of complex dynamics in the Natural Sciences". We thank Martin Böhme for comments on a first draft of this manuscript.

References

BARTH, E., DORR, M., BÖHME, M., GEGENFURTNER, K. R., AND MARTINETZ, T. 2006. Guiding the mind's eye: improving communication and vision by external control of the scanpath. In *Human Vision and Electronic Imaging*, B. E. Rogowitz, T. N. Pappas, and S. J. Daly, Eds., vol. 6057 of *Proc. SPIE*. Invited contribution for a special session on Eye Movements, Visual Search, and Attention: a Tribute to Larry Stark.

BÖHME, M., DORR, M., MARTINETZ, T., AND BARTH, E. 2006. Gaze-contingent temporal filtering of video. In *Proceedings of Eye Tracking Research & Applications (ETRA)*, 109–115.

BURT, P. J., AND ADELSON, E. H. 1983. The Laplacian pyramid as a compact image code. *IEEE Transactions on Communications 31*, 4, 532–540.

DELABARRE, E. B. 1898. A method of recording eye-movements. *American Journal of Psychology 9*, 4, 572–574.

DORR, M., BÖHME, M., MARTINETZ, T., AND BARTH, E. 2005. Visibility of temporal blur on a gaze-contingent display. In *APGV 2005 ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, 33–36.

DORR, M., VIG, E., GEGENFURTNER, K. R., MARTINETZ, T., AND BARTH, E. 2008. Eye movement modelling and gaze guidance. In *Fourth International Workshop on Human-Computer Conversation*.

DUCHOWSKI, A. T., BATE, D., STRINGFELLOW, P., THAKUR, K., MELLOY, B. J., AND GRAMOPADHYE, A. K. 2009. On spatiochromatic visual sensitivity and peripheral color LOD management. *ACM Transactions on Applied Perception 6*, 2, 1–18.

GEISLER, W. S., AND PERRY, J. S. 2002. Real-time simulation of arbitrary visual fields. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, New York, NY, USA, 83–87.

HEMINGHOUS, J., AND DUCHOWSKI, A. T. 2006. iComp: a tool for scanpath visualization and comparison. In *APGV '06: Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, ACM, New York, NY, USA, 152–152.

JARODZKA, H., SCHEITER, K., GERJETS, P., VAN GOG, T., AND DORR, M. 2009. How to convey perceptual skills by displaying experts' gaze data. In *Proceedings of the 31st Annual Conference of the Cognitive Science Society*, Austin, TX: Cognitive Science Society, N. A. Taatgen and H. van Rijn, Eds., 2920–2925.

JARODZKA, H., SCHEITER, K., GERJETS, P., AND VAN GOG, T. 2010. In the eyes of the beholder: How experts and novices interpret dynamic stimuli. *Journal of Learning and Instruction 20*, 2, 146–154.

JARODZKA, H., VAN GOG, T., DORR, M., SCHEITER, K., AND GERJETS, P. 2010. Guiding attention guides thought, but what about learning? Eye movements in modeling examples. (submitted).

LANKFORD, C. 2000. Gazetracker: software designed to facilitate eye movement analysis. In *ETRA '00: Proceedings of the 2000 symposium on Eye tracking research & applications*, ACM, New York, NY, USA, 51–55.

LESGOLD, A., RUBINSON, H., FELTOVICH, P., GLASER, R., KLOPFER, D., AND WANG, Y. 1988. Expertise in a complex skill: diagnosing X-ray pictures. In *The nature of expertise*, M. T. H. Chi, R. Glaser, and M. Farr, Eds. Hillsdale, NJ: Erlbaum, 311–342.

NIKOLOV, S. G., NEWMAN, T. D., BULL, D. R., CANAGARAJAH, C. N., JONES, M. G., AND GILCHRIST, I. D. 2004. Gaze-contingent display using texture mapping and OpenGL: system and applications. In *Eye Tracking Research & Applications (ETRA)*, 11–18.

RAMLOLL, R., TREPAGNIER, C., SEBRECHTS, M., AND BEEDASY, J. 2004. Gaze data visualization tools: Opportunities and challenges. In *IV '04: Proceedings of the Information Visualisation, Eighth International Conference*, IEEE Computer Society, Washington, DC, USA, 173–180.

STONE, L. S., MILES, F. A., AND BANKS, M. S. 2003. Linking eye movements and perception. *Journal of Vision 3*, 11 (11), i–iii.

UNDERWOOD, G., CHAPMAN, P., BROCKLEHURST, N., UNDERWOOD, J., AND CRUNDALL, D. 2003. Visual attention while driving: Sequences of eye fixations made by experienced and novice drivers. *Ergonomics 46*, 629–646.

UZ, K. M., VETTERLI, M., AND LEGALL, D. J. 1991. Interpolative multiresolution coding of advanced television with compatible subchannels. *IEEE Transactions on Circuits and Systems for Video Technology 1*, 1, 86–99.

VAN GOG, T., JARODZKA, H., SCHEITER, K., GERJETS, P., AND PAAS, F. 2009. Effects of attention guidance during example study by showing students the models' eye movements. *Computers in Human Behavior 25*, 785–791.

VELICHKOVSKY, B., POMPLUN, M., AND RIESER, J. 1996. Attention and communication: Eye-movement-based research paradigms. In *Visual Attention and Cognition*, W. H. Zangemeister, H. S. Stiehl, and C. Freksa, Eds. Amsterdam, Netherlands: Elsevier Science, 125–54.

VIG, E., DORR, M., AND BARTH, E. 2009. Efficient visual coding and the predictability of eye movements on natural movies. *Spatial Vision 22*, 5, 397–408.

ŠPAKOV, O., AND MINIOTAS, D. 2007. Visualization of eye gaze data using heat maps. *Electronica & Electrical Engineering 2*, 55–58.

ŠPAKOV, O., AND RÄIHÄ, K.-J. 2008. KiEV: A tool for visualization of reading and writing processes in translation of text. In *ETRA '08: Proceedings of the 2008 symposium on Eye tracking research & applications*, ACM, New York, NY, USA, 107–110.

WOODING, D. S. 2002. Fixation maps: Quantifying eye-movement traces. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, ACM, New York, NY, USA, 31–36.

YARBUS, A. L. 1967. *Eye Movements and Vision*. Plenum Press, New York.