Adaptive Perception with Locally-Adaptable Sensor Array

Róbert Wagner, Ákos Zarándy and Tamás Roska

***Abstract*— In this paper we propose some biologically motivated local image sensor adaptation methods. The special feature of these methods are, that they change the integration time of the imager on the pixel level. Using these methods, low dynamic range integration type CMOS sensors will be able to perceive high dynamic range scenes, and at the same time, compress their dynamic range by keeping high contrast and without introducing non-existing edges.**

***Index Terms*—locally adaptive image perception, high dynamic range image sensors, bio-inspired visual sensor adaptation.**

I. Introduction

EQUALLY illuminated scenes have 2-3 log orders (8-10 bits) dynamics only, while unequally illuminated scenes might easily bear with 5 decades (17 bits) of dynamics. Perception of these high dynamic range scenes is an easy job for our eyes, however it is a great challenge for traditional image sensors. If we want to capture these high dynamic range scenes, an obvious idea is the selection of a logarithmic imager. Though it can cover the high dynamic range, but as an exchange, the image will be noisy and low contrast.

If we better quality images (low noise, high contrast), we have to choose integration type linear imagers. However, their dynamic range is typically 8-10 bits, which fact predestinates their failure (saturation) in case of high dynamic range scenes. The only way to bridge this problem is to take a few snapshots with different exposure times. In this case, all parts of the scene will be depicted in one of the snapshots in good quality. However this works only for still scenes. Moreover, a new problem rises: how to combine these snapshots into a single good quality image. Trivial solution is the linear combination, however it leads to an image with 16-18 bits dynamic, which cannot be displayed on any nowadays-existing devices. (Displays can visualize images with 6-8 bits dynamic only.)

Our goal is to overcome these weaknesses of the integrating type linear sensors, namely to modify them on a way, that in their new form, they will be able to capture the whole large dynamic scene in a single snapshot, resulting a low noise, high contrast image with 8-10 bits dynamic range. This requires a tricky dynamic compression, similar to what our retina does.

The key issue here is the accuracy, which depends on the operation point of the individual sensory elements. Our basic idea is to locally tune the operation point of a photo sensor matrix on a way by controlling its integration time on the individual sensory element level. This approach is very similar to the operation method of our retina.

However, two problems arises immediately: (i) how to get a sensor with this local integration type controlling property, and (ii), how to control the integration time to avoid spurious edges and other artifacts. The answer for the first question is the following: recently research activities were focused to develop locally tunable sensors in some research laboratories (e.g. IMSE-CNM Seville Spain, MTA-SzTAKI, Budapest, Hungary, [1-3]). An answer for the second question is this paper, where we introduce some biologically motivated algorithms, which generates the integration time map for the locally controllable photo sensor matrix.

The biological background is presented in Section II. The algorithms and the experimental results are shown in Section III and IV. Finally, we compare our results with other solutions in Section V.

II. Biological Motivations

A. Retinal Adaptation

The retina consists of several layers, from which mostly two layers take part in the dark- and light-adaptation. These are the photoreceptor and the horizontal cell layers, which constitute the outer plexiform layers (OPL) with the bipolar cells.. Here we consider only this part of the retina.

Our cone photoreceptors perform logarithmic compression of the input. The cones can sense a dynamic range of 9-10 log orders, and reduce it to 2-3 log orders [6]. This is achieved by shifting the response curve to the appropriate dynamic range [6]. This means that the photoreceptor adapts to the changing illumination conditions. An example can be seen in Fig.1

Analogical and Neural Computing Laboratory, Hungarian Academy of Sciences, and the Jedlik Laboratories, Faculty of Information Technology, Pázmány Péter. Catholic University, Budapest; email: [wagnerr,zarandy,roska]@sztaki.hu; web: lab.analogic.sztaki.hu

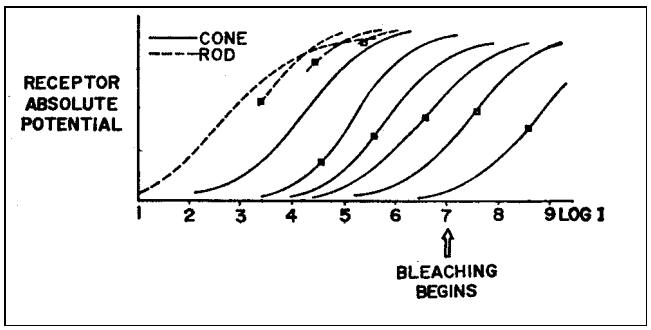


Fig. 1. Necturus cone (and rod) response curves (response at one adaptation state to different stimuli). The adaptation state was achieved by showing a given background levels - indicated by squares- to the receptors for sufficient long time to adapt. One can see the shifting of the curves. The whole dynamic range is smaller than that of humans (10 log order). (Adopted from [3])

If the incoming light changes suddenly the receptor gives a high response and may saturate. At steady illumination state the receptor adapts to the changed brightness conditions and does not saturate. This implies that temporal changes are enhanced by the photoreceptors.

The second level of adaptation is the OPL, where the horizontal cells perform a spatial-temporal low-pass filtering [7,8]. Then, this is subtracted from the photoreceptor output. Hence the spatial-temporal low-passed component is suppressed and the dynamic range is reduced by 1 log order.

These modifications imply the following:

- The suppression of the spatial-temporal low-passed component eliminates the less important information. The observer is rather interested in the local differences (high-passed component) [9,11], because these represent the objects. The suppressed low-passed component relates to the illumination [10,11], which caused the large dynamics.
- According to the rules of diffuse reflection the perceived light is proportional with the illumination. So to reduce the effects of the spatial-temporal variant illumination we need a division by the illumination level. Due to the logarithmic compression at the receptor level this can be realized as a subtraction of the horizontal cell output.

Hence the retina enhances the information about the object contours, and suppresses the less important illumination components. It reacts to sudden brightness changes and later adapts to these conditions after they have stabilized [6,9].

B. Biological and Artificial Sensors

In this subsection, we show, that using the integration-time as a locally controllable parameter we can obtain retina-like image enhancement. It is obvious that the integration time is a multiplicative factor (neglecting dark current [4]). This means that integrating *n* times longer we get *n* times larger pixel value, unless the sensor saturates. Hence changing the integration time of the photo sensor its characteristics is shifted along the intensity axis, which is similar to the adaptation course of the cones. Integration type linear image sensor and ideal cone response curves with different integration time/adaptation stages can be seen in Fig. 2.

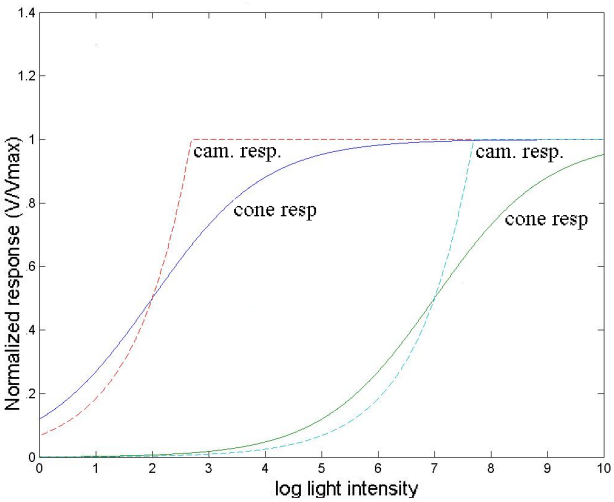


Fig. 2. Camera responses (dashed lines) and ideal cone responses (solid lines) at different adaptation states. Vmax denotes the maximum receptor response. Changing the integration time the camera response curve obeys similar shifting as the cone curves, thus the camera response curve can be considered as an approximation of the cone response curve.

As mentioned the integration time parameter is a multiplicative factor. Hence by properly choosing its value we can eliminate (or suppress) the multiplicative effect of the illumination, as in the case of the retina. To achieve this we need a proper algorithm, which adjusts the integration time based on local image statistics. We discuss this question in the following Sections IV. But before that, in Section III we explain our experimental framework.

III. MIMICKING THE OPERATION OF A LOCALLY ADJUSTABLE PHOTO SENSOR MATRIX

In this Section, we explain how we mimic the operation of a locally adjustable photo sensor matrix. Since we do not have this special sensor in our hand, mimicking is essential to be able to test and evaluate our algorithms.

We capture a sequence of grayscale pictures of the same high dynamic scene with different global integration time parameters (T_1, T_2, T_3, ….T_n). Then, we interpolate each pixel value of the image. Given an integration time T at a pixel, we wish to derive the intensity V as the mimicked sensor would have perceived. We choose the two pictures with the nearest integration time values T_i and T_{i+1} on a way, that $T_i > T > T_{i+1}$. V_1 and V_2 are the two intensity values measured in the same pixel position with T_1 and T_2 integration time respectively. There is a linear relationship between the perceived intensity and the integration time. Hence target value V can be computed as the weighted average of V_1 and V_2:

$$V = \frac{(T - T_{i+1})V_i + (T_i - T)V_{i+1}}{T_i - T_{i+1}} \qquad (1)$$

Using equation 1 we can synthesize the captured image based on the picture series of different integration times. This assembling technique is similar to the multiple capturing method in [4]. For capturing the image sequence, we used a Dalsa 1M28 camera [14]. The images were 8bit-depth, and the integration time was changed between 0.1 and 230 ms.

In order to make experiments with movies we have to make a 4D image flow, where we have a picture series of different integration times in place of a frame. So we have to take several pictures at a given moment. If taken a picture series of the motion scene, the scene would be easily altered between snapshots with different integration time. To avoid this problem we made a series of static scenes instead of a motion scene. This can be viewed as if we have frozen the motion at some discrete moments to create the picture series (of different integration time).

IV. Integration Time Adjustment Algorithms

In section II we learned how the response curve of the cones is shifted to the perceived light intensity (Fig.1), and the low-pass (DC) component is suppressed in the OPL. Both of these can be achieved by adjusting the integration time so, that the DC component (local average) becomes the half of the maximum value ($V_{max}/2$). This means that the camera response curve should be shifted depending on the local brightness level. This eliminates the intra scene DC differences, because the DC values are driven to $V_{max}/2$ in each locations. The maximal slope of the photoreceptor responses curve at the local average is obtained if this value is projected to $V_{max}/2$ [15]. Similarly if the local average is projected to $V_{max}/2$ the camera curve is still steep around this level (see Fig.2.). Hence we can have great responses for the local differences, the high-passed component.

The low-pass (DC) component is the local average around each pixel. This can be computed by using diffusion operator. To proper adjust the integration time we present two different methods. The first is based on intra frame processing done during the image capturing. It assumes very special architecture, dedicated for this calculation.

The second is based on inter frame processing. It is an iterative method, where the iteration is done through the captured frames. This method is simpler from computational and hence implementation point of view, but as an exchange, the adaptation to a new illumination condition, similarly to our retina, takes time.

A. Processing During Integration

There is an obvious method if we have an ultra fast computer, which can do calculation during the integration time. Focal plane processor arrays [18] with appropriate instruction set are expected to be able to do this sometimes in the future.

The goal is to achieve, that the local average of the responses becomes $V_{max}/2$. To obtain this we have to permanently calculate the local average in each locations, and when it reaches $V_{max}/2$, we should stop the integration.

This method keeps the local average always at $V_{max}/2$. Since this is an intra frame method, it responds immediately to any motion and illumination changes in the scene. It is similar to Carver Mead's method [8], but while he computed the local average and subtracted it from the actual pixel value, we control the integration time locally. We expect that local integration time tuning leads to higher dynamic range and accuracy than pure local DC level subtraction, because, all of our sensory elements will be working in an operating point, where they are very sensitive.

B. Iterative Adjustment

In this subsection, we present two iterative methods. In Fig. 3 we can see a general flowchart of these methods. As a first step, we define an initial integration time map T_0. Then, we repeat the followings in each iterative step. At the n-th iteration we capture an image with the integration time map T_n and get the result image V_n. Then, we perform diffusion operation on both V_n and T_n. Based on the diffused images we compute the integration time map T_{n+1} for the next snapshot. This is the adaptation phase where we change the variable parameter T_n in order to adapt to the scene.

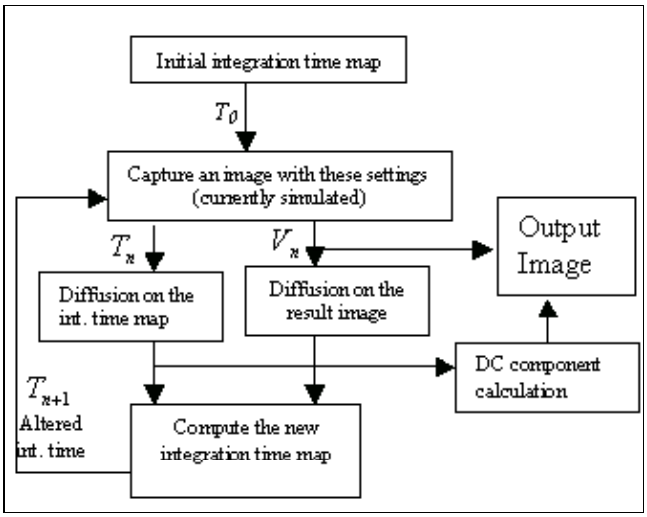


Fig. 3. Flowchart of the iterative adjustment of the integration time. The DC component can be computed based on the integration time values, and it is added to the resulting image.

The next two subsections deal with the computing of the new integration time map (integration time adjustment), while the third explains the DC level restoration method. This last one is needed, because the proposed adaptive sensing algorithms are bleaching out the DC component, and enhances the edges.

1) Multiplicative Adjustment

At the n-th iteration we have the integration time setting ($T_{n,i,j}$) for pixel (i,j). With these settings we obtained the measured value $V_{n,i,j}$. The goal is to adjust the integration time at this pixel so, that the local average of $V_{(n+1)}$ should be $V_{max}/2$. We compute the local average of V_n with diffusion (see Fig.3.). Denote the local average for pixel (i,j) as $\overline{V_{n,i,j}}$.

Denote the local average of the integration time for pixel (i,j) as $\overline{T_{n,i,j}}$. Change the integration time at pixel (*i*,*j*) as follows (later referenced as multiplicative formula):

$$T_{(n+1),i,j} = \overline{T_{n,i,j}} \frac{V_{\max}}{2\overline{V_{n,i,j}}} \tag{2}$$

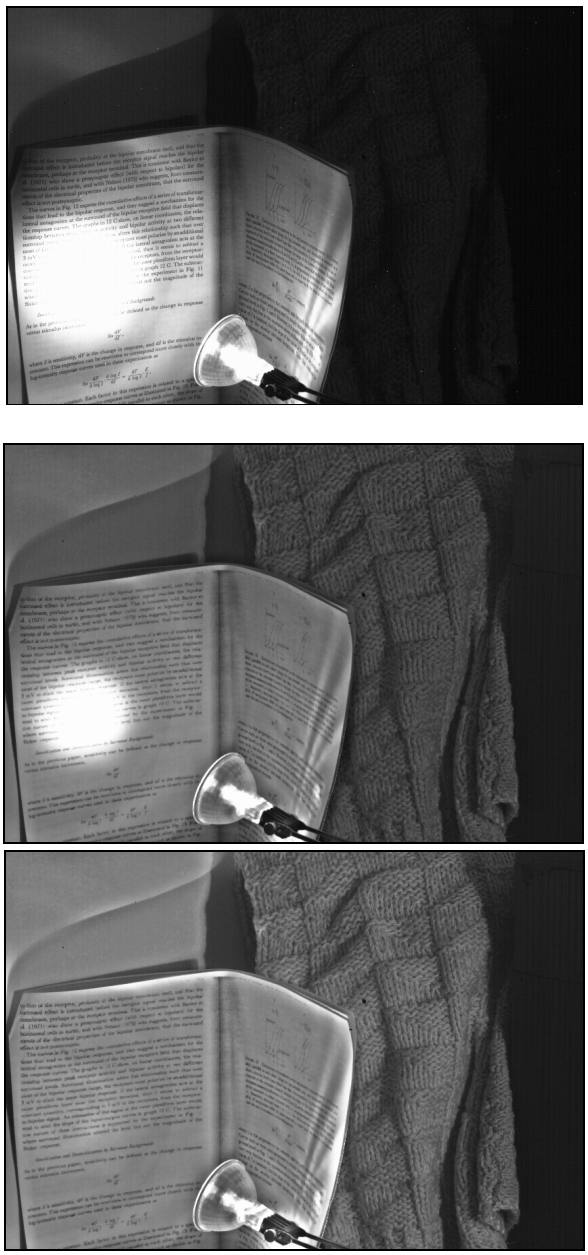
The received intensity $V_{i,j}$ is proportional to the integration time $T_{i,j}$. So the heuristic is obvious, if we obtained $\overline{V_{n,i,j}}$ with $T_{n,i,j}$, than we have to multiply $T_{n,i,j}$ with $V_{\max} \Big/ 2\overline{V_{n,i,j}}$ to get $V_{max}/2$. The heuristic assume that the neighboring pixels are altered in the same way. This may hold for most cases, because both $\overline{V_{n,i,j}}$ and $\overline{T_{n,i,j}}$ are low passed components.

Instead of $T_{n,i,j}$ we compute with $\overline{T_{n,i,j}}$ (local average), to avoid sudden changes in the integration time map.

The free parameter of the method is the neighborhood radius (neighRad), on which we build the local average. Using the diffusion operation on CNN architecture the analog transient execution time corresponds with neighborhood radius [13].

Fig. 4. shows some results of this method. After the first alteration of the integration time map, the dark areas are enhanced, and the saturated white zone is reduced. The saturated white area is not completely eliminated, because due to saturation the correction of Eq.2. does not diminish the integration time enough. After sufficient iterations (Fig.4.c.) we reach an adapted state where the local average is nearly everywhere $V_{max}/2$. On the right side it was not enhanced more for practical reasons because the integration time of the camera could not be set higher. Apart from this we can observe that the strongly illuminated areas and the dark background are driven to the same brightness level.

The multiplicative adjustment provides us a fast adaptation. On the other hand we have to take into account some implementation aspects as well. The division in Eq. 2 may not be easily implemented on CMOS CNN-UM chip, hence we propose the additive adjustment.



a.) b.) c.)

Fig. 4. Picture (a) image captured at global 16ms integration time, which was the original image. Picture (b) and (c), the result of the multiplicative correction after the 1st and 4th iteration respectively. *neighRad*=20

2) Additive

Using the notations of the previous section, we compute the followings instead of Eq.2:

$$T_{(n+1),i,j} = \overline{T_{n,i,j}} + \left(\frac{1}{2} - \frac{\overline{V_{n,i,j}}}{V_{\max}}\right)\overline{T_{n,i,j}} = \\ = \left[1 + \left(\frac{1}{2} - \frac{\overline{V_{n,i,j}}}{V_{\max}}\right)\right]\overline{T_{n,i,j}} \tag{3}$$

This expression can be computed without division. The

division with V_{max} can be made as a multiplication, since V_{max} is a constant. To better understand the adjustment we may compute the following value: Depending on $\overline{V_{n,i,j}}$ we multiply $\overline{T_{n,i,j}}$ with a factor in either Eq. 2 or 3. Denote factor $M_{n,i,j}$. Plotting this multiplication factor ($M_{n,i,j}$) ve $\overline{V_{n,i,j}}$, we can see how the integration time is enhance different brightness levels. Fig. 5. shows this integration t multiplication factor ($M_{n,i,j}$) versus the local brightness leve

Based on the first plot of Fig. 5. we can make a compari between the additive and multiplicative enhancement. I important to note that the local average at value $V_{max}/$ unchanged (multiplication by 1). So this is an equilibr point of the system. Local averages greater or smaller than are reducedor enhanced respectively. The amount diminution of bright values is quite similar (see Fig.6.). Tl is a great difference in the enhancement of the dark values Fig. 6.). These are far less enhanced using the addi correction (see Fig. 7.).

In order to accelerate the adaptation of dark regions, proposed an iterated-additive adjustment, further referre *fast additive correction*. We can alter the integration time *n* times based on the same local average values $\left(\overline{V_{n,i,j}}\right)$. is the same as if we multiplied $\overline{T_{n,i,j}}$ with $\left(M_{n,i,j}\right)^n$ inst of $M_{n,i,j}$. The resulting multiplication factor is plotted in 6., on the second subplot with dashed line. We can see that, the enhancement of the dark values became better. On the other hand, the integration times for the bright values are too much diminished. Hence we need a threshold operation [13], and for the pixels brighter than $V_{max}/2$ we keep the result of the original additive correction. The data-flow of the fast additive correction can be seen in Fig. 8.

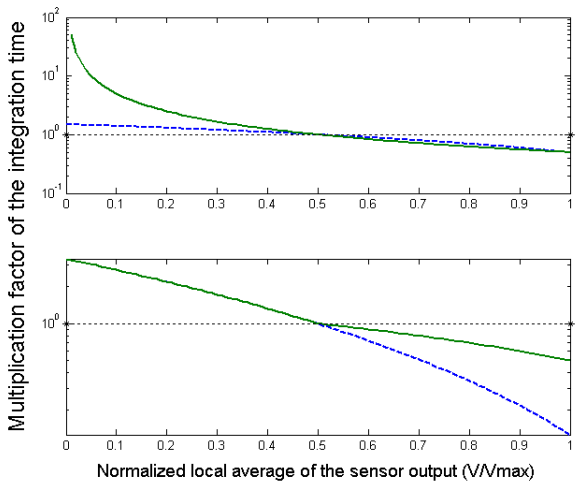


Fig. 5. The factors with which we multiply the integration time at different local average values. The upper plot shows the multiplicative (solid curve) and the additive (dashed line) correction. The second plot shows the curves of the fast additive corrections. In this case, we took the third power of the factors, (dashed curve). This curve works fine for the dark values, but original additive worked better for the bright values. Since that, we combined these two curves in a way, that we took the third power of the factors only when they were larger than 1 (solid curve). The dim doted line is the 1 level, which means that the integration time is kept unchanged.

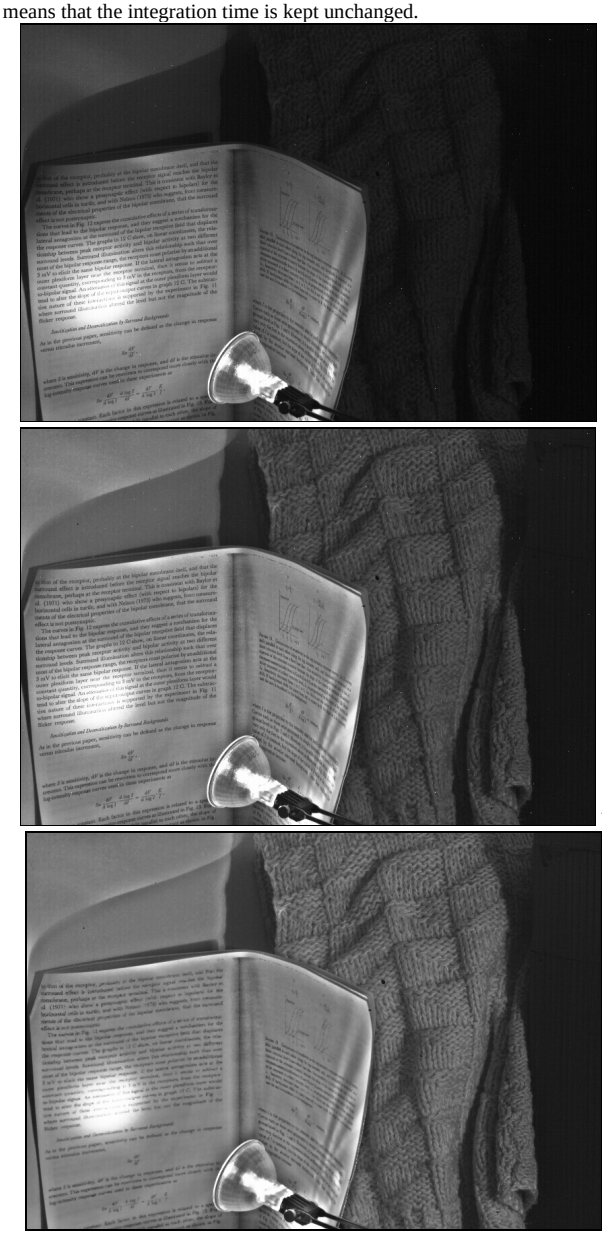


Fig. 6. Result images after the second capture a.) additive, b.) fast additive adjustment and c.) multiplicative adjustment. We can see, that at bright areas all methods adapt in a similar way. On the darker regions the iterative method is faster (b.), than the non-iterative additive (a.) (the dark region are more enhanced). After further iterations the methods yield similar results as in Fig. 4.

Fig. 6. shows adjustment results of the same initial scene as Fig. 5. As it can be seen, the fast additive adjustment adapts faster at dark areas than the additive adjustment. The additional iterations do not notably retard the system. On the CNN-UM architecture these operations can be fulfilled faster then the capturing of the images. In the following sections this fast additive method will be used for computing the alteration of the integration time. This method requires a second tunable parameter the iteration number.

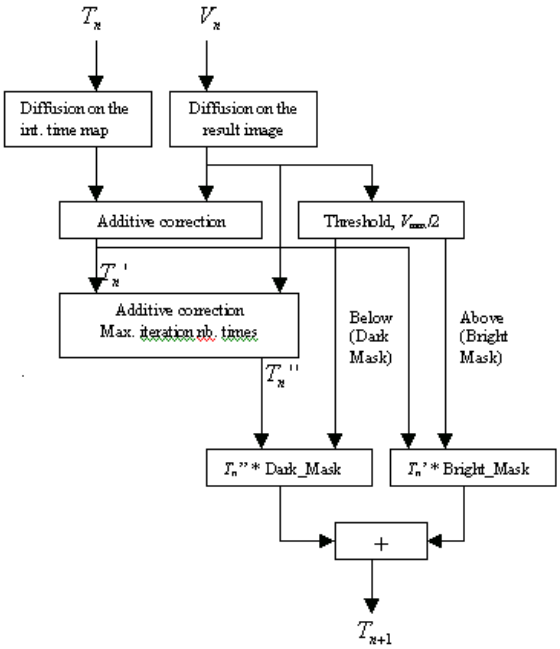


Fig. 7. Data-flow of the fast additive way of computing the integration Detailed subpart of Fig. 3.

3) Restoring DC level

The methods presented so far enhance the high-p component of the images: edges, local differences. eliminate the effect of the differences in the local average driving all the averages to the half of the maximum respo This is very advantageous for example in the case of the b part of the scene, where we will have equal local avera (Fig. 4. and 6.). On the other hand the illumination differer are totally disappeared. Furthermore the intensity differer between some objects are also disappeared, because th were in the same order of spatial frequency band as illumination. E.g. intensity difference between the pullo and the book.

Perceptually it might be better to preserve the component on a way to add a downscale component of i the resulting image. The integration time map carries information about the local averages.

Having adapted the integration time the local averag projected to the half of the maximum value:

$$\overline{V_{i,j}} * T_{i,j} = \frac{V_{\max}}{2} k = const, \tag{4}$$

k is an appropriate constant. This means:

$$\overline{V_{i,j}} \sim \frac{1}{T_{i,j}} \tag{5}$$

Thus in the visual system every information is logarithmically compressed, we can obtain the DC level as following:

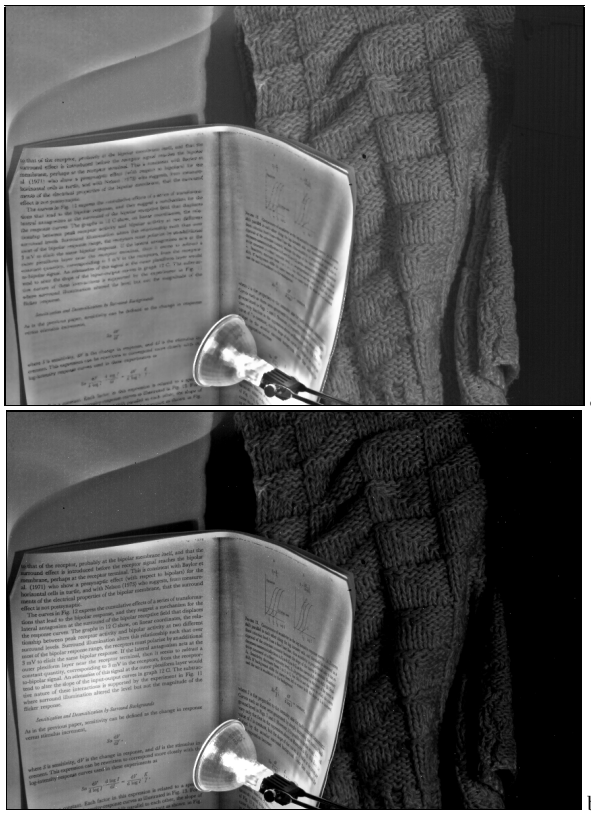
$$V_{DC,i,j} = -\log(T_{i,j}) \tag{6}$$

After that this value should be normalized between $-V_{max}/2$ and $V_{max}/2$. With these values at the darkest areas the local average will be shifted from $V_{max}/2$ to 0 and at the brightest to V_{max}. Using the DC level restoring we have a third parameter to adjust: c_{DC} , which is a multiplicative parameter. $c_{DC} \in [0,1]$

$$V_{DC,i,j} = \lfloor normalize(-\log(T_{i,j})) \rfloor * c_{DC} \tag{8}$$

Fig. 8. shows result with and without DC component.

Using CNN-UM architecture the DC level can be computed by interpolating the logarithmic curve with some circuit elements.



a.)

b.)

Fig. 8. Result images after the 8th adjustment.. (a) without DC component, (b) DC component added. c_{DC}=0.5

Our proposed method has three free parameters:

neighRad the neighborhood radius of the low-pass filtering (usually set one tenth of the image size). Using diffusion template operation the execution time corresponds to this parameter.

Iteration number iteration number of the additive adjustment. (we set 3 in the simulation)

c_{DC} Ratio of the DC component. Usually 0.7 was set, when DC component was needed.

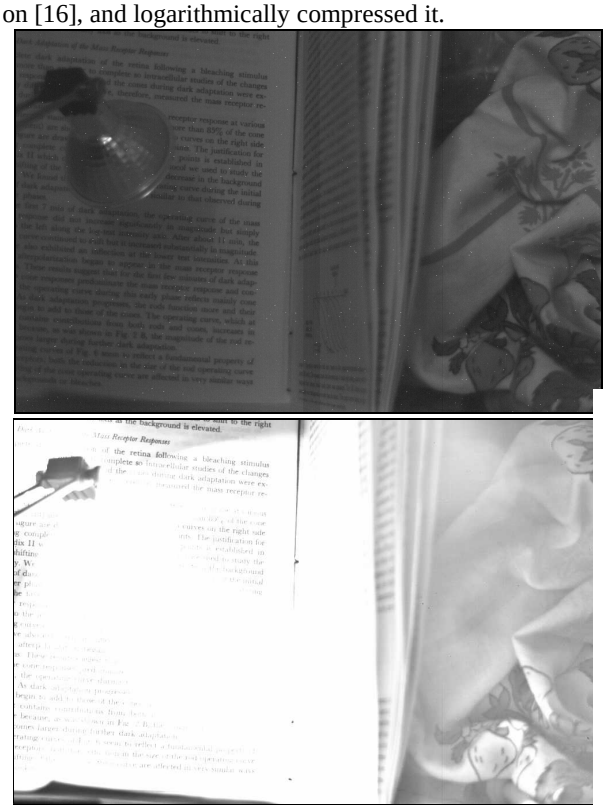
4) Dynamic Results

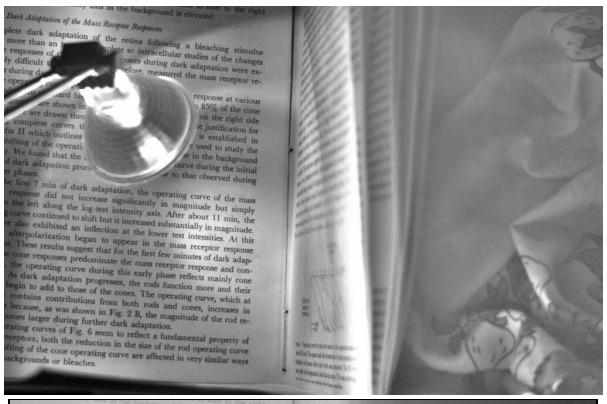
We have applied our methods to dynamic image sequence.

In this case, the input scene, captured in each iteration s changes in time. Hence the adaptation mechanism has follow this permanent changing. According to experiments, if the changes are not very fast on the scene, output image sequence adaptation is much quicker, and method provides good quality images most of the time. show here an experience in which first the scene unaltered, only the strong illuminator was switched on, t the position of the strong illuminator was changed.

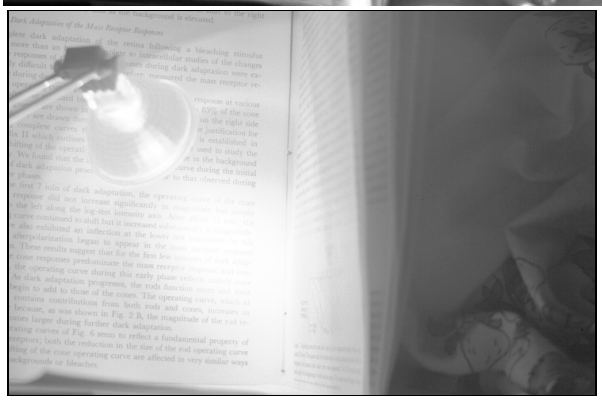
The assembling was based on an integration time computed with the threshold-iterated additive adjustm Some frames of the dynamic adjustment can be seen in Fig and 10. The parameter settings were the followi *neighRad*=0, *iteration number* =3 and c_{DC}=0. The comp image flow can be viewed in [17].

To have a good comparison of the adaptation we compu a logarithmic-compressed image from the image series. F the series we computed a high-dynamic (16bit) image ba on [16], and logarithmically compressed it.



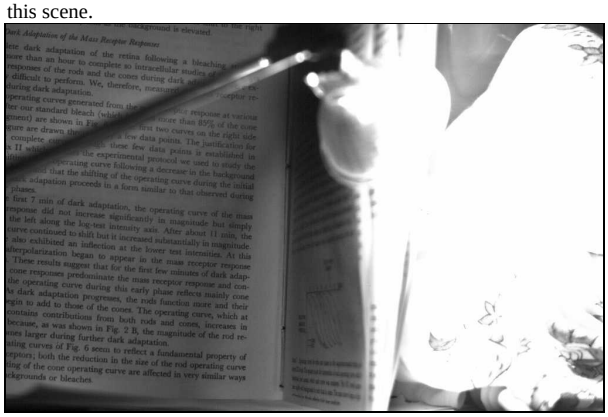


c.)

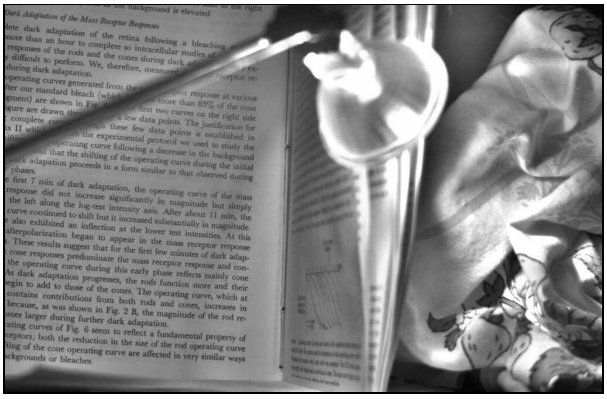


d.)

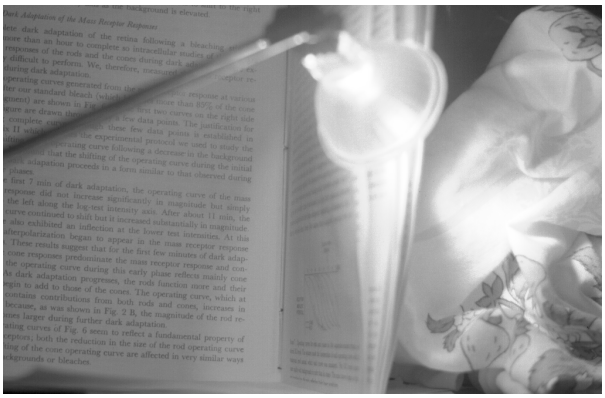
Fig. 9. The illumination was switched on during the first 6 frame of 60 frames. Between frames 6 and 14, the illumination was unaltered, to provide time for adaptation. a.) The original, equal illuminated scene with global integration time. b.) The switched on illumination, the 6th frame. Picture c.) the adapted state, the 14th frame. Picture d.) the logarithmic compression of this scene.



a.)



b.)



c.)

Fig. 10. The illuminator was movied between frame 14 and 22. a.) shows the 22nd frame. In this frame, the illuminator shifted to the right significantly. After this, we did not alter the scene until the 31st frame to provide time for the adaptation (b). c.) is the logarithmic compression of this scene.

Regarding Fig. 9. and 10. we can see the dynamic behavior of our method. Having a change in the illumination the algorithm needs an adaptation time, to calculate the new integration time map. Before having adapted to the new conditions we get dark values where we diminished the illumination (Fig.10.a.) the book) and vice versa (Fig.10.a.) the right side of the image). This can be viewed as a memory; the system remembering to its past. This means that we have a temporal high-pass filtering effect as well. This is similar to the retinal processing, where the spatial-temporal low-pass component is suppressed and the high-pass components enhanced (see section II).

V. Comparison to other high dynamic range Sensors

A. *Logarithmic sensors*

In [4] high dynamic range sensing methods are compared. Our method corresponds to the locally adjustable integrating methods (Multiple captures, Spatially varying exposure). These methods yielded better signal-noise ratio (SNR) than the other non-integrating sensors. The Multiple Capturing method takes images of different integration time (similar way as in our experiments). The Spatially Varying Exposure method requires more sensors with different capturing parameter for one pixel. The first requires more time the latter looses spatial resolution.

Compared to logarithmic sensors, based on [4] we can state, that the integrating sensors have better SNR than the logarithmic sensors. Furthermore we can observe the contrast relationship of the two methods.

We may compute the sensitivity defined as the slope of the response curves [4]. This value is proportional response to a certain input difference. Assume an ideal logarithmic sensor, which spans the log-dynamic range linearly (see Fig. 11). The adaptive method shifts the response curve to the local dynamic range by adjusting the integration time. Thus adapted to the local dynamic range it has a greater slope than the slope of the logarithmic sensor's response curve.

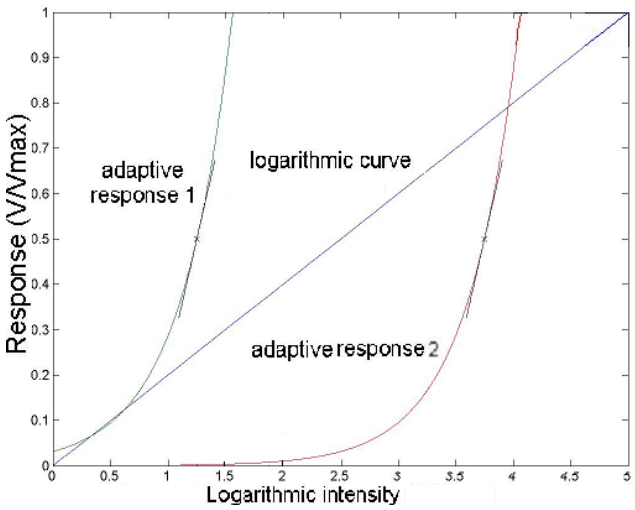


Fig. 11. Comparison of the logarithmic and the variable integration time sensors. The ideal logarithmic sensor spans the whole intensity range and saturates at the end (diagonal line). The exponential curves show adaptive integration time responses at different local averages (indicated by x). At the local averages the slopes of the tangential are indicated.

This means that the same intensity difference around the local average generates a greater response, and yields a greater contrast using local adaptation (compare Fig. 9.c. and d.).

The fact that the slope of the adaptive integration time sensor's response curve is larger than that of the logarithmic sensor can be deduced formally as well:

The response of the ideal logarithmic sensor versus the log-intensity:

$$V_l[\log(I)] = \frac{V_{\max}}{\log(D)}\log(I) \tag{9}$$

where D is the dynamic range spanned by the logarithmic sensor and log(.) means the natural logarithm.

The slope of the curve, the sensitivity as used in [4]:

$$S_l = \left.\frac{\partial V_l(I)}{\partial I}\right|_{I=I_0} = \left.\frac{\partial V_l(\log(I))}{\partial(\log(I))}\frac{\partial \log(I)}{\partial I}\right|_{I=I_0} = \frac{V_{\max}}{\log(D) I_0} \tag{10}$$

We compute the sensitivity at $I=I_0$, where I_0 is the local average, around which we compare the sensitivity with the adaptive integration time characteristics'. Thus integrating sensors have linear response, at the adapted state it can be computed following (neglecting dark current):

$$V_i(I) = \frac{I V_{\max}}{2 I_0} \tag{11}$$

The sensitivity computed as before:

$$S_i = \left.\frac{\partial V_i(I)}{\partial I}\right|_{I=I_0} = \frac{V_{\max}}{2 I_0} \tag{12}$$

The ratio between the two sensitivities

$$r_{i,l} = \frac{S_i}{S_l} = \frac{\log(D)}{2} = \frac{\log(2)\log_2(D)}{2} \qquad (13)$$

This ratio specifies how much steeper are the adaptive integration time curves than an ideal logarithmic curve (see Fig. 11). E.g. in case of a 16-bit scene ($\log_2(D)=16$): $r_{i,l} = 5{,}54$, which means that using adaptive integration time method the same intensity difference evokes a 5,54 times greater response. So theoretically a logarithmic sensor needs 3-bit deeper accuracy to code the same information.

Real logarithmic sensors do not certainly span their response range within the input range. Due to the fact, that they would span it on a wider range (which makes the slope flatter) the LinLog characteristics is applied [14]. The initial part of such a response curve is linear and after a specified value it is logarithmic. In this case the log response curve (Fig. 11) is not linear and at some locations the slope is flatter again.

B. *Other Adaptive Sensors*

In this subsection we deal with pixel level adaptive algorithms and implemented sensors which are capable of perceiving high dynamic range scenes. Some of these also use spatially varying exposure parameters.

In [19] a hardware implementation is presented, where the capacitance of a pixel is reset if the pixel saturates. The sensors' output needs post processing: the output of the pixels where the integration was stopped is needed to be normalized with the integration time.

[20] uses integration of the photocurrent, until the pixel intensity reaches a given threshold level. After reaching the level it sends an impulse (spike),resets the sensor and continues intergrating. Thus the frequency of the spikes is proportional to the intensity on the scene. The whole dynamic range is coded in the duration of the intervals between subsequent spikes, so the sensor requires high temporal sensitivity.

In our method, the integration time of a pixel corresponds to the adaptation state of the cones. These two have a shifting effect along the log-intensity axis. We do not retrieve the high dynamic range scene as [19,20], but perform an adaptation and a high-dynamic range compression already at the sensor level. By dividing the perceived intensity with the integration time we could also compute the original high dynamic-range brightness of the scene.

The method presented in [21] uses several pictures captured with different global exposure parameters (e.g. integration time). It builds the result image by summarizing the pictures. The key points are the values of the exposure parameters. These are set to maximize the slope of the resulting response curve compared to a desired curve (e.g. linear curve, see [21]). Our method aims to maximize the slope of the response curve location dependent, which is achieved by adjusting the integration time. We take only one image of a scene to retrieve the result, so we do not loose time, for capturing several images.

[22] gives another solution without the loss of temporal resolution. It uses regulary varying exposures, e.g. different exposure values in a 2x2 box, and repeat this box on the whole sensor array. Thus the neighboring cells will have different settings. After taking an image, the pixels whose exposure parameter was not appropriate for the region are omitted (e.g. saturation, dim response). Then after a normalization it performs an interpolation on the usable pixels to obtain the result high-dynamic range image. In our method we adjust the integration time according to the local average. Thus - with the exception of high spatial temporal brightness gradients – we have the perceived intensity in the middle portion of the response curve, (and we do not have to omit points).

VI. Conclusions

In this paper we presented our work on a local-adjustment algorithm for the future locally adjustable sensor chip. This retina inspired algorithm enhances the spatial-temporal high-pass component of the high dynamic range scene. Beside that it suppresses the low-pass component thus reduces the dynamic range of the scene. We have also presented some experimental results showing the perception of static and time variant scenes. Compared the method to other high dynamic range sensing systems, our system will be adequate in perceiving these scenes. Using our method, each kind of photodetectors' dynamic range can be extended.

Acknowledgment

This research was founded by the Grant of the National Science Fund of Hungary (OTKA), the multidisciplinary doctoral school at the Faculty of Information Technology of the Pázmány P. Catholic University and the Office of Naval Research (ONR) Grant No. N0001-4021-0884 and the European Community (Grant No. IST-2001-38097 LOCUST).

References

[1] T.Roska and L.O.Chua, "The CNN Universal Machine: An Analogic Array Computer", IEEE Trans. Circuits and Systems, Ser.II., vol. 40, pp. 163-173, 1993

[2] T. Roska, "Computer-Sensors: Spatial-Temporal Computers for Analog Array Signals, Dynamically Integrated with Sensors", *Journal of VLSI Signal Processing*, Vol.23, pp.221-237, 1999

[3] T. Roska and Á. Zarándy: "Proactive Adaptive Cellular Sensory-Computer Architecture via extending the CNN Universal Machine", to be published on the *ECCTD '03 European Conference on Circuit Theory and Design*, 1 - 4 September 2003, Kraków, Poland

[4] A. El Gamal: "High Dynamic Range Image Sensors", Tutorial at *International Solid-State Circuits Conference*, February 2002, Available: http://www-isl.stanford.edu/~abbas/group/papers_and_pub/isscc02_tutorial.pdf

[5] B. Roska and F.S. Werblin: "Vertical Interactions Across Ten Parallel Stacked Representations in the Mammalian Retina." *Nature* **410:** 583-587 (2001)

[6] R. A. Norman and F. S. Werblin: "Control of Retinal Sensitivity: I. Light and Dark Adaptation of Vertebrate Rods and Cones", *The Journal of General Physiology*, Vol.63, pp.37-61, 197

[7] F. S. Werblin: “Control of Retinal Sensitivity: II. Lateral Interactions at the Outer Plexiform Layer”, *The Journal of General Physiology*, Vol.63, pp.62-87, 1974

[8] Carver A. Mead and M. A. Mahowald: ”A silicon model of early visual processing”, *Neural Networks*, Vol. 1, Issue 1, pp. 91-97 (1988)

[9] L. Spillmann and J. S. Werner: “Visual Perception – The Neurophysiological Foundations”, 5 / III., Academic Press, Inc., San Diego, California, 1990

[10] V. Brajovic: ”A model for reflectance perception in vision”, *Bioengineered and Bioinspired Systems, Proceedings of SPIE*, Vol. 5119 (2003)

[11] J. E. Dowling: “Neurons and Networks, An Introduction to Neuroscience”, Retinal Processing of Visual Information, The Belknap Press of Harvard University Press, Cambridge, 1992

[12] Homepages about diffuse and specular reflection. Available:
http://scienceworld.wolfram.com/physics/Reflection.html
http://www.siggraph.org/education/materials/HyperGraph/illumin/diffuse_reflection.htm

[13] T. Roska, L. Kék, L. Nemes, Á. Zarándy and P. Szolgay (ed), “CNN Software Library (Templates and Algorithms), Version 7.3”, *Analogical and Neural Computing Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SzTAKI)*, DNS-CADET-15, Budapest, 1999

[14] DALSTAR 1M28-SA, CMOS Area Scan Cameras - Camera User’s Manual. Available:
http://www.ctc-g.co.jp/~hts/dalsa/pdf/1M28_DALSTAR-SA.pdf

[15] W. H. A. Beaudot: “Sensory coding in the vertebrate retina: towards an adaptive control of visual sensitivity”, Published in *Network: Computation in Neural Systems* as a Symposium Paper, Vol.7(2), pp.317-323, May 1996

[16] D. Yang, A. El Gamal, B. Fowler, and H. Tian, “A 640x512 CMOS image sensor with ultra-wide dynamic range floating-point pixel-level ADC”, *IEEE. J. Solid-State Circuits*, vol. 34., no. 12, pp 1821-1834, Dec. 1999.

[17] Image flows of the dynamic adjustment are available at the following sites:
Uncompressed: http://digitus.itk.ppke.hu/~wagner/Add.avi
Compressed: http://digitus.itk.ppke.hu/~wagner/Add_c.avi
Compressed logarithmic flow:
http://digitus.itk.ppke.hu/~wagner/Log_c.avi

[18] G. Linan: “ACE16K: an Advanced Focal-Plane Analog Programmable Array Processor”, *ESSCIRC 2001 Presentations 27th European Solid-State Circuits Conference*, Villach, Austria, 18-20 September 2001

[19] T. Hamamoto and K. Aizawa: “A Computational Image Sensor with Adaptive Pixel-Based Integration Time”, *IEEE Journal of Solid State Circuits*, Vol. 36. no. 4 April. 2001

[20] E. Culurciello, R. Etienne-Cummings, and K. Boahen: “A Biomorphic Digital Image Sensor”, *IEEE Journal of Solid-StateCircuits, Vol. 38., No. 2. February 2003*

[21] M. D. Grossberg and S. K. Nayar: “High Dynamic Range from Multiple Images: Which Exposures to Combine?”, *Int. Proc. ICCV Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*, Nice, France, October 2003.

[22] S. K. Nayar and T. Mitsunaga: “High Dynamic Range Imaging: Spatially Varying Pixel Exposures”, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, Hilton Head Island, South Carolina, June 2000.