System of automatic video stream images evaluation and preprocessing for ADAS

A A Egorchev
Kazan Federal University, 18 Kremlyovskaya str, Kazan, 420008, Russia
E-mail eanton090@gmail.com

Abstract. In this article, the algorithm of preliminary filtering of images of a video stream based on assessment and the subsequent correction of such qualitative characteristics of images as contrast, noise level, sharpness and existence of illumination is considered.

Keywords: image preprocessing; image filtering; noise reduction; unsharp mask; histogram correction; ADAS

1. Introduction
One of the most important tasks of ADAS systems is to classify the environment using computer vision algorithms. Using unprocessed video streams and images in the algorithms often fails to achieve sufficient accuracy because of the quality of the visual image settings (low sharpness, contrast, noise, etc.), and also increases the processing time due to the presence of large amounts of unused data (e.g., color channels of the image or high resolution image), which makes impossible the use such approaches in systems operating in real-time.

To assess the quality of the input image, the criteria corresponding to the intervals of feature values, as well as functions for evaluating each criterion for the image flow will be defined. The functions for evaluation of criteria were tested on various types of video (with different levels of sharpness, contrast, time of day, varying degree of noise).

The video stream of images, previously converted to 256 grayscale mode (8bit grayscale) using colorimetric conversion with preservation of brightness, is used as input data [1]. (The input data is a video stream of images previously converted to 256 grayscale (8bit grayscale) mode using a colorimetric transformation with preservation of brightness [1].) The content of the images is the results of shooting video cameras installed inside the cabin on the windshield of various cars.

In this field of science multiple authors offer a number of solutions for related problems. Ralph Gross and Vladimir Brajovic propose a new image preprocessing algorithm that compensates for illumination variations in images. From a single brightness image, the algorithm first estimates the illumination field and then compensates for it to mostly recover the scene reflectance. Unlike previously proposed approaches for illumination compensation, our algorithm does not require any training steps, knowledge of 3D models or reflective surface models.[2] The authors use an image preprocessing solution to improve the quality for facial recognition. Jones M. in their scientific work described the operational system for the precision preprocessing of imagery. The operational system performs three basic functions: image data acquisition; amplitude preprocessing, including such functions as interchannel registration, line-phase correction, and fine gain adjustment; and navigation,
involving the production of accurate image deformation models [3]. Siddhartha Bhattacharyya in the above article presents a brief survey of the trends in color image enhancement and segmentation: methods in this direction involve Markov random models, vector directional filters and statistical mixture models like Gaussian and Dirichlet mixtures, the non-classical approaches comprising the neuro-fuzzy-genetic paradigm or its variants are bestowed with features for real time applications [4].

Solving specific tasks in narrowly defined areas requires different approaches to image preprocessing, since the criteria for image preprocessing differ. In particular, a system of automatic video stream images evaluation and preprocessing was developed for ADAS. The research results presented in this article are unique and were obtained as a result of many practical experiments.

2. Criteria for evaluating images of video stream

2.1. Noise level
Noise estimation is important in many image processing and analysis algorithms. Quantifying the noise component makes it possible to adapt the preprocessing algorithms to the amount of noise instead of using fixed threshold values.

The model for noisy images is as follows (formula 1):

\[ I(x,y) = f(x,y) + n(x,y), \]

where \( f \) is the "perfect" image, \( n \) is the noise, \( I \) is the observed image.

One of the main problems of noise estimation is the measurement of deviations \( I \) from \( f \), which may contain significant structural information, such as edges and texture features.

[5] presents a fast algorithm of assessment of image noise dispersion. It is used as method for noise determination in our task. In the offered method the operator with a zero average value, which is almost insensitive to the image structure is used. The dispersion of the output values is taken as an estimate of the noise variance. Since the structural elements of images, such as edges, correspond to the differential components of the second order, the noise estimation must be resistant to the Laplacian of the image.

The advantage of this method is that it uses the Laplace operator, which is insensitive to the image structure, and its result depends only on the noise level in the image.

When using the ESTIMATE_NOISE function that implements this method, the values of the \( NL \) noise level in the image belong to intervals \( NL \in (0,3) \cup [3, +\infty) \). The value of \( NL = 3 \) is considered a threshold: images with the value of \( NL \) from 3 and above are considered noisy and require additional processing.

The image is pre-translated into Grayscale. At high noise levels above \( NL = 3 \), the median noise reduction operation is applied, at lower noise levels, the image does not need additional processing.

The median filter is a nonlinear digital filtering technique that is often used to remove noise from an image or signal.

Median filtering is one of the ways to smooth the image, equally with linear Gaussian filtering. All smoothing methods are effective at removing noise in smooth areas without structural features, but adversely affect the edges. When choosing smoothing filtering algorithms, it is important to be guided by the degree to which the filter allows you to preserve edges that bear significant visual information about the image. In this respect, for small or medium levels at a given fixed window size, the median filter is significantly better than the Gaussian blurring.

2.2. Level of sharpness
The sharpness of the image describes the subjective visibility of details, the degree of clarity of the border between the two sections of the photographic image that have received different exposures. The sharpness value is related to the amplitude of the brightness derivative in spatial variables.

In the most general sense, the sharpness of the image \( S \) at any point can be taken as the brightness gradient \( I \) (formula 2).

\[ S = \nabla I. \]
Thus, the sharpness of the image is a vector field.

For the input image using the ESTIMATE_SHARPNESS function, the sharpness of the $SL$ is measured (method [6], based on the calculation of the image gradient modulus). $SL$ values are less than 2 correspond to the image of low sharpness, from 2 to 4.5 – the average, more than 4.5 – high sharpness (see examples in figure 1).

![Figure 1](image.png)

Figure 1. (a) low sharpness image ($SL = 1.7$), (b) high sharpness image ($SL = 6.3$)

Images with a high level of sharpness do not need to be pre-processed. For images with sharpness values lower than 4.5, an unsharp masking filter is applied.

The unsharp masking filter applies a blur to a copy of the original image. If, when comparing the mask with the original, the differences exceed a certain threshold, the image is subtracted. The threshold setting is necessary to avoid amplifying unwanted details such as noise in a digital photo or grain on a film.

A typical scheme of applying an unsharp masking filter is as follows (formula 3):

$$sharpened = original + (original - blurred_r) \times amount$$

(3)

where original is the input image, blurred$_r$ is the image with the blur filter applied (usually Gaussian) with radius $r$, which affects the degree of blurriness of the image copy and the width of the zone in which the change in tone will appear, amount is an Unsharp masking parameter that determines how much the areas of the image that are on the boundaries of the contour are darkened or lightened.

When using the unsharp masking filter, the threshold parameter is also used, which determines the minimum difference in tone at which the unsharp masking is accomplished. For a noisy or grainy original, it is set to a higher value to prevent these unwanted details from underlining. Pre-noise reduction allows this parameter to be fixed ($threshold = 0$).

For sharpness values 4.5 and above, no filtering is applied to the image. If $SL \in (0, 2)$ an unsharp masking filter is applied with the parameters $amount = 0.5$, $radius = 10$. If $SL \in [2, 4.5)$ an unsharp masking filter is applied with parameters $amount = 0.5$, $radius = 20$.

2.3. The presence of illumination

The illumination on the image is characterized by the presence of overexposed regions in the area of excessive sunlight on the matrix, the presence of sharp shadows, as well as areas of low contrast, which has an extremely negative impact on the subsequent recognition of the image objects.

The FLASH_TEST function is used to determine the presence of $FL$ illumination.

The illumination on the image is characterized by the presence of overexposed regions in the area of excessive sunlight on the matrix, the presence of sharp shadows, as well as areas of low contrast, which has an extremely negative impact on the subsequent recognition of the image objects.

The FLASH_TEST function is used to determine the presence of $FL$ illumination.

Since the input image was converted to grayscale mode calorimetrically, the value of each pixel retains the illumination parameters. To determine the presence of illuminated parts in the image, 3 windows ($W_1, W_2, W_3$) are selected, located in two corners and center of the image. Then the average pixel value (brightness) in each window is measured: $Y_{av}(W_1), Y_{av}(W_2), Y_{av}(W_3)$ (see examples in figure 2). If threshold $th = 200$ is exceeded in at least one of the windows, the values of $Y_{av}(W_i)$ are compared. If there is a difference in brightness values of more than 100%, then there is illumination and the value of $FL$ is set to 1, otherwise $FL = 0$. 
Figure 2. An image with illumination ($FL = 1$). Brightness levels $Y_{ab}$ of the image illumination

If there is a illumination in the image ($FL = 1$), an adaptive histogram alignment filter is applied to the image (see examples in figure 3).

Histogram alignment [4] is used in image processing to increase the contrast level. This method typically increases the global contrast of many images, especially when pixel-by-pixel image contrast data is represented by similar values. By using histogram alignment, the intensity values can be better distributed across the histogram. This allows you to increase the contrast of areas of low local contrast ratio. Histogram alignment allows you to efficiently distribute the most common intensity values in an image.

The disadvantage of the method is its non-selectivity and the fact that its application can increase the contrast of background noise of the image. Therefore, noise reduction is used as a pre-processing for noisy images.

Adaptive histogram equalization (AHE) [8] is an image processing technique used to improve global contrast. Ordinary histogram equalization uses the same transformation derived from the image histogram. This method works well when the distribution of pixel values is the same throughout the entire image. However, when an image contains areas that are significantly lighter or darker than the rest of the image, the contrast in these regions when applied to the image of a regular histogram alignment will not be sufficiently enhanced.

The adaptive method calculates several histograms, each corresponding to a separate part of the image, and uses them to redistribute brightness values of the image. Therefore, it is suitable for improving local contrast and edge refinement in all areas of the image.

Contrast Limited Adaptive histogram equalization (CLAHE) [9] is a version of adaptive histogram equalization in which contrast enhancement is limited to avoid the occurrence or amplification noise in the image. In the case of the illuminated image, adaptive histogram alignment helps to smooth out the contrast of the image in both over- and underexposed areas, and to get rid of sharp shadows.

Figure 3. (a) Image with the presence of illumination on the left ($FL = 1$) in the grayscale mode (b) Image with an adaptive histogram alignment filter
2.4. Contrast Level

Contrast measures the difference in brightness or color that makes objects in an image distinguishable. The contrast level is related to the intensity gradient. In addition, contrast can be defined locally or globally. When an image consists of textured areas, it is more appropriate to perform local image analysis and determine local contrast. It is well known that the level of local contrast can be increased by calculating the differences between the intensity value in each pixel of the image and its surrounding pixels and the subsequent increasing these differences [10].

For the input image, the contrast level $CL$ is measured using the method of A. Beghdadi and A. L. Negrate [11]. This algorithm is based on the Gordon’s method [7], which directly determines the contrast function and the function of improving the contrast of the image without involving the histogram of the gray levels of the input image. To improve the operation of the method in conditions of high noise level and image detail, algorithms for edge selection are implemented in this method.

The algorithm is implemented in the ESTIMATE_SHARPNESS function.

For the current image, the contrast is measured and recorded in $CL$. At values of $CL$ from 0 to 70 the image is considered low contrast. When $CL > 70$, the image is high contrast. See examples in figure 4.

![Figure 4](image)

Figure 4. (a) low contrast image ($CL = 32$), (b) high contrast image ($CL = 110$)

For high-contrast images $CL \geq 70$ filtering is not applied (figure 5(a)). At $CL \in (0, 70)$ the filter CLAHE is applied (figure 5(b)).

![Figure 5](image)

Figure 5. (a) low contrast image ($CL = 32$) in grayscale mode (b) image with histogram alignment filter applied to it ($CL = 67$)

3. Algorithm for automatic evaluation and filtration of images

Figure 6 shows the algorithm of automatic filtering of the image in a video stream. In order to avoid excessive computational costs, as well as to optimize the operating time of the algorithm, the frequency of application is determined for each criteria evaluation function, based on the nature of the image properties change in the video stream, as well as the requirements of subsequent recognition algorithms. Table 1 shows the order and frequency of application of criteria evaluation functions.
### Table 1. Frequency of application of functions.

| Function          | Frequency of application (number of applications per second) |
|-------------------|-------------------------------------------------------------|
| ESTIMATE_ NOISE   | 6                                                           |
| ESTIMATE_ SHARPNESS| 3                                                           |
| FLASH_TEST        | 18                                                          |
| ESTIMATE_ CONTRAST| 3                                                           |

Figure 6. Algorithm of automatic evaluation and filtering
4. Software implementation

Jetson TX2, an NVidia industrial product, is used for the automatic evaluation and image preprocessing system. The characteristics of the Jetson TX2 are listed in Table 2.

| Feature Specification | Jetson TX2 |
|------------------------|------------|
| Processor              | NVIDIA Pascal architecture with 256 CUDA cores |
| Computing Micromodule  | HPM Denver 2/2 ARM A57/2 |
| Video                  | Encoding 4K x 2K 60Hz (HEVC) |
|                        | 4K x 2K 60Hz decoding |
|                        | (12 bit support) |
| Memory                 | 8 GB memory |
| Display                | 2x DSI, 2x DP 1.2 / HDMI 2.0 / eDP 1.4 |
| CSI                    | Up to 6 cameras (2 channels) |
|                        | CSI2 D-PHY 1.2 (2.5 Гб/s per channel) |
| PCIE                   | Gen 2 | 1x4 + 1x1 or 2x1 + 1x2 |
| Data storage           | 32 Гб eMMC, SDIO, SATA |
| Other                  | CAN, UART, SPI, I2C, I2S, GPIOs |
| USB                    | USB 3.0 + USB 2.0 |
| Connection             | 1 Gigabit Ethernet, 802.11ac WLAN, Bluetooth |

The location of the Jetson TX2 computer in the logic of the ADAS system is shown in Figure 7.

Running applications on this product involves running resource-intensive parts of the program on the GPU. In connection with this requirement, computer vision algorithms were developed using NVIDIA’s CUDA software-hardware parallel computing architecture, which the program to increase its speed by using computational micromodules.

The algorithmic base of the developed system was implemented using OpenCV and VisionWorks machine vision libraries based on CUDA technology. The generalized block diagram of software modules interaction is presented in figure 8.

To ensure real-time operation of the software, the calculation of the image characteristics was distributed over time intervals. During the processing of one frame, one image characteristic is calculated, the image is filtered at each step based on the last measurements.

The operating time of the algorithms on Jetson TX2 is presented in table 3. The tests were carried out on images in HD resolution.
Table 3. Jetson TX2 Computer Specifications.

| Algorithm Name | Operating time, ms |
|----------------|-------------------|
| Median         | 14                |
| Unsharp        | 23                |
| CLAHE          | 12                |
| Estimate noise | 15                |
| Estimate sharpness | 40            |
| Estimate contrast | 50            |
| Flash test     | 14                |

An example of the results of the algorithm is presented in Figure 9.

5. Conclusion
The considered algorithm for automatic filtering of video stream images uses evaluation of characteristics for video stream images and can be used to improve the quality of detection and classification of various types of objects in ADAS systems.

6. Acknowledgment
The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.

References
[1] Poyton C A 1998 Rehabilitation of gamma *Proc. SPIE 3299 Human Vision and Electronic Imaging III* ed B E Rogowitz and T N Pappas (San Jose: SPIE) pp 232-249 [https://doi.org/10.1117/12.320126](https://doi.org/10.1117/12.320126)
[2] Gross R and Brajovic V 2003 An image preprocessing algorithm for illumination invariant face recognition *Lecture Notes in Computer Science* **2688** 10-18
[3] Jones M, Lauque P and Haemmer R 1980 An introduction to the meteosat image preprocessing and image navigation system *ESA Journal* **4** 211-226
[4] Bhattacharyya S 2011 A brief survey of color image preprocessing and segmentation techniques *Journal of Pattern Recognition Research* **6** 120-129
[5] Immerkaer J 1996 Fast noise variance estimation Computer Vision and Image Understanding **64** 300-302
[6] Birdal T 2019 Sharpness estimation from image gradients *MATLAB Central File Exchange* [https://www.mathworks.com/matlabcentral/fileexchange/32397-sharpness-estimation-from-image-gradients](https://www.mathworks.com/matlabcentral/fileexchange/32397-sharpness-estimation-from-image-gradients)
[7] Gordon R and Rangayyan R M 1984 Feature enhancement of film mammograms using fixed and adaptive neighborhoods *Applied Optics* **23** 560-564
[8] Hummel R 1977 Image enhancement by histogram transformation *Computer Vision, Graphics and Image Processing* **6** 184-195
[9] Pizer S M, Amburn E P and Austin J D 1987 Adaptive histogram equalization and its variations
   Computer Vision, Graphics and Image Processing 6 355-368
[10] Kim. V and Yaroslavsky L P 1986 Rank algorithms for picture processing Computer Vision,
    Graphics and Image Processing 6 234-258
[11] Beghdadi A and Negrate A L 1989 Contrast enhancement technique based on local detection of
    edges Computer Vision, Graphics and Image Processing 46 162-174