The technology of correction of dynamic distortions on mobile devices

E F Fatkhutdinova[1] and V A Fursov[1,2]

[1]Samara National Research University, Moskovskoe Shosse 34, Samara, Russia, 443086
[2]Image Processing Systems Institute - Branch of the Federal Scientific Research Centre “Crystallography and Photonics” of Russian Academy of Sciences, Molodogvardeyskaya str. 151, Samara, Russia, 443001

Abstract. We propose the technology of image correction in mobile devices based on the use of a parametric FIR filter. The filter is constructed using a frequency response specified in the form of a parametric family of centrally symmetric frequency characteristics. A model of a one-dimensional frequency response is used in the form of segments of three functions: parabola, constant, and exponential function. Within the framework of the proposed filter model, a mobile application has been created that implements two tuning schemes: identification of the filter parameters from the reference image and adjustment without a reference. Besides, it is possible to vary the frequency response parameter characterizing the mid-range interval, which provides additional possibilities for adjusting the quality of the recovery. An example of processing test images is given.

1. Introduction

Statistics of recent years show that the volume of traffic of mobile devices on the Internet is growing rapidly. A mobile application is a program that works on tablet PCs and smartphones. The number of smartphone owners around the world is growing every day. Music, working with photos, social networks - the most popular types of applications among smartphone owners [1]. One of the most popular functions of mobile devices is image registration[2-3]. Mass use of this function is associated with the ability to quickly register images in unexpected and unique situations. In this case, often you have to deal with problems due to defocusing and/or blurring (when the object being recorded is rapidly moving relative to the camera) [4]. Distortions such as blurring and defocusing are usually called dynamic. The task of correcting dynamic distortions in images recorded by mobile devices using the digital image processing algorithms implemented directly in the mobile device itself is extremely relevant.

Known classical approaches to the construction of filters for image processing are inverse filtering and Wiener filtering. When constructing an inverse filter, one often has to face the fact that the inverse operator does not exist or the corresponding transfer function has poles close to zero [5]. At the same time, noise is emphasized on the restored image. In the Wiener filter, this problem is overcome by taking into account in the filter transfer function the frequency characteristics of noise [6]. However, in practice these characteristics are often absent, and the synthesis of the optimal Wiener filter becomes a serious problem. Unfortunately, other known methods of constructing filters, in one form or another, also face these problems [5], [6]. Therefore, often filters for the correction of dynamic distortions are built by parametric identification in the class of filters with finite impulse response (FIR filters). In [7], such a technology was considered, based on the use of a parametric model of the frequency response in the form of segments of quadratic and exponential functions. In this work, two variants of the technology were considered: identifying the parameters of the impulse response by use cases and setting the filter by indirect characteristics without a reference image.

In [8], the model of the frequency response from segments of quadratic and exponential functions was generalized, in particular, the region of medium frequencies was expanded due to the adding of a frequency response constant interval. On model examples it was shown that this modification improves the quality of image reconstruction in comparison with the previous realization of the quadratic exponential FIR filter. However, the implementation of a filter with an extended frequency response region requires, albeit insignificantly, an increase in computational resources. In this paper, we present the results of work on developing a mobile application that implements both of the above options.

The work is structured as follows. Section 1 describes the methods and algorithms for filter adjustment, both for the reference image and for the absence of a reference. In Section 2, the libraries used for constructing the mobile application, pseudo- code and user interface are explained. Section 3 provides examples of processing test images in a mobile device.

2. Method and algorithms

An FIR filter with a radially symmetric real frequency response [5, 6] is constructed with a supporting region D in the form $N \times N$ - a square with the center at the point $k_1 = 0,\ k_2 = 0$. Assuming that the central sample of the impulse response *h(0,0)* of the reference region D is at point n_1, n_2 readings of the restored image $y(n_1, n_2)$ [5] can be represented as:

$$y(n_1,n_2) = \sum_{k_1=-\frac{N-1}{2}}^{\frac{N-1}{2}} \sum_{k_2=-\frac{N-1}{2}}^{\frac{N-1}{2}} h[r(k_1,k_2)]x(n_1-k_1, n_2-k_2), \qquad (1)$$

where $r(k_1,k_2) = \sqrt{k_1^2 + k_2^2}$, and $h[r(k_1,k_2)]$ – counts of the one-dimensional impulse response defined on a set of circles with radii $r = r(k_1,k_2),\ k_1, k_2 \in D$.

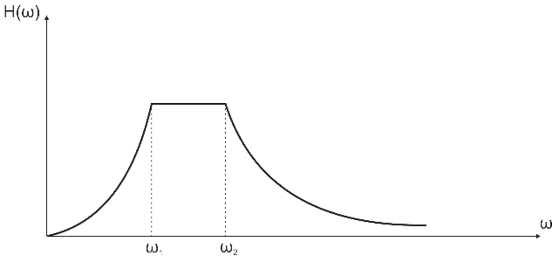


Figure 1. A typical graph of the spectrum of the GSE filter.

The one-dimensional frequency response function is used for all values $0 \le \omega < \infty$ in the form of three consecutive segments: parabola, constant and exponential function:

$$S(\omega) = \begin{cases} a\omega^2, in\ 0 \le \omega < \omega_1, \\ A = const = a\omega_1^2, in\ \omega_1 \le \omega \le \omega_2, \\ e^{-c\omega}, in\ \omega \ge \omega_2, \end{cases} \qquad (2)$$

$$S(\omega_2) = a\omega_1^2 = e^{-c\omega_2}.$$

Filter, corresponding to the described frequency response, hereinafter referred to as a generalized square-law exponential filter (Generalized Square-Exponential) or briefly a GSE-filter. The graph of this function is shown in figure 1.

The impulse response corresponding to this spectral characteristic, by virtue of the radial symmetry property, is obtained as a function of the spatial parameter r inverse Fourier transform:

$$h(r)=\frac{1}{\pi}\operatorname{Re}\int_0^{\infty}S(\omega)e^{j\omega r}d\omega=\frac{1}{\pi}\operatorname{Re}\left\{\int_0^{\omega_1}a\omega^2e^{j\omega r}d\omega+\int_{\omega_1}^{\omega_2}Ae^{j\omega r}d\omega+\int_{\omega_2}^{\infty}e^{-c\omega}e^{j\omega r}d\omega\right\}= \tag{3}$$

$$=\frac{e^{-c\alpha\omega_1}}{\pi}\left\{\frac{\sin(\omega_1 r)}{r}\sin(\omega_1 r)+\frac{2\cos(\omega_1 r)}{\omega_1 r^2}-\frac{2\sin(\omega_1 r)}{\omega_1^2 r^3}+\frac{\sin(\alpha\omega_1 r)-\sin(\omega_1 r)}{r}+\frac{c\cos(\alpha\omega_1 r)-r\sin(\alpha\omega_1 r)}{c^2+r^2}\right\}$$

(we excluded a and ω_2, taking into account (2) substitution $a = {e^{-c\omega_2}}/{\omega_1^2}$ and take $\omega_2 = \alpha\omega_1$).

It is easy to see that when $\alpha = 1$ impulse response (3) coincides with the impulse response of the quadratic-exponential filter, which we considered in the previous paper [7]. In this paper we will build a mobile application in which the user can set this parameter (or the range of this parameter change) taking into account subjective requirements for the quality of recovery and available computing resources.

The counts of the two-dimensional impulse response, as in [8], are determined by discretizing the continuous function (3) for all directions corresponding to all samples of the reference region. For each sample (points k_1, k_2) of the support region in the ratio (3), the argument $r = r(k_1, k_2) = \sqrt{k_1^2 + k_2^2}$. Since at $r=0$ in (3) there is uncertainty, the value of the central reading is calculated as the sum of all samples except for the central one:

$$h(0,0) = \sum h(k_1, k_2),\ \forall k_1, k_2 \in D, k_1, k_2 \neq 0. \tag{4}$$

Then, all the samples in the reference area are normalized so that the requirement of maintaining the average brightness level of the processed image is fulfilled:

$$\sum h(k_1, k_2) = 1,\ \forall k_1, k_2 \in D. \tag{5}$$

The filter adjustment algorithm is constructed in the following sequence of steps:

1. Assigning the initial estimates of the parameters $\hat{\omega}_1, \hat{c}, \hat{\alpha}$ and criterion $Q_k(\hat{\omega}_1, \hat{c}, \hat{\alpha})$ (with $k = 0$).

2. Calculation of pulse response samples for all points of the reference region $D(n_1, n_2)$ with the use of ratios (3), (4) and the normalization of all samples, satisfying (5).

3. Processing of the distorted image and calculation of the quality criterion $Q_k(\hat{\omega}_1, \hat{c}, \hat{\alpha})$.

4. If the $Q_k(\hat{\omega}_1, \hat{c}, \hat{\alpha}) > Q_{k-1}(\hat{\omega}_1, \hat{c}, \hat{\alpha})$, estimates $\hat{\omega}_1, \hat{c}, \hat{\alpha}$ save, otherwise, according to some rule, a new variant of estimates is formed and the transition to step 2 is performed. If all estimates from the range of admissible values are "viewed" is an output.

Based on the described technology, it is possible to adjust the filter parameters both from the reference image and without the reference. When setting up a benchmark as a criterion $Q_k(\hat{\omega}_1, \hat{c}, \hat{\alpha})$ proximity of the restored image to the standard is used indicator:

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right), \tag{6}$$

where *MAX* – is the maximum value received by the image pixel, and *MSE* – is the root-mean-square error (MSE) characterizing the proximity of the restored image to the reference image.

In the filter adjustment technology without a reference image, an MSE is calculated between the recovered and the original distorted images. In this case, in contrast to the setting according to the etalon, the improvement in the quality of the restored image is accompanied by a decrease in the PSNR (6). As the PSNR value is reduced, the quality of the resulting image can be either higher or lower, an additional condition is used: increasing the variance of the processed image. This requirement provides an increase in the average contrast, which is usually observed with an increase in the sharpness of the image. In addition, a restriction on the minimum allowable value of PSNR is introduced, since at its small values, significant distortions, characterized by high contrast, are possible.

Thus, checking the fulfillment of the conditions on the k-th iteration $Q_k(\hat{\omega}_1, \hat{c}, \hat{\alpha}) > Q_{k-1}(\hat{\omega}_1, \hat{c}, \hat{\alpha})$ on step 4 of the described technology, it reduces to verifying that the following conditions are met:

$$\begin{gathered} PSNR\left(\hat{\omega}_k, \hat{c}_k,\ \hat{\alpha}_k)\right) < PSNR\left(\hat{\omega}_{k-1}, \hat{c}_{k-1},\ \hat{\alpha}_{k-1})\right) \\ D\left(\hat{\omega}_k, \hat{c}_k,\ \hat{\alpha}_k\right) > D\left(\hat{\omega}_k, \hat{c}_k,\ \hat{\alpha}_k\right), \\ PSNR\left(\hat{\omega}_k, \hat{c}_k,\ \hat{\alpha}_k\right) > PSNR_{доп}. \end{gathered} \tag{7}$$

Here $PSNR(\widehat{\omega}_k, \hat{c}_k, \hat{\alpha}_k)$ value indicator (6) calculated on the k - th iteration of the image reconstructed with filter parameters $\widehat{\omega}_k, \hat{c}_k, \hat{\alpha}_k$ when compared with the reference (when tuning to the etalon) or the original distorted (when configured without reference); $D(\widehat{\omega}_k, \hat{c}_k, \hat{\alpha}_k)$ – the dispersion of the reconstructed image, and $PSNR_{\text{доп}}$ – the minimum permissible value of the exponent (6).

3. Development of mobile application

The development of any mobile image processing application is divided into 2 parts: the development of a filter and the development of the main part. For the first part - the implementation of the image processing algorithm - the OpenCV library was used, a special version for the Android platform. OpenCV is a crossplatform open source library that is free for commercial and academic use. OpenCV contains algorithms for interpreting images, determining optical distortions, determining the similarity, analyzing the movement of an object, determining the shape of an object, and much more, written in C / C ++ and using parallelism and multi-core for the tasks performed.

To use the OpenCV library on the Android platform there is also a ready-made Android set NDK - tool that allows to realize part of the Android app compiled programming languages such as C and C ++ and contains a library to manage activities and access to various physical components of the device. Android NDK is integrated with tools from a set of components for software development (Android SDK), as well as with the integrated development environment of Android Studio [9-11].

Thus, there are 2 ways to use the OpenCV library on the Android platform:

1. OpenCV Java API + Android SDK: functionality is written in the Java language;
2. OpenCV native interface + Android NDK: functionality is written in C ++.

We will use the first way.

The main library modules used are:

- module Core - contains basic operations, including arithmetic;
- module Imgproc - responsible for image processing;
- module Utils - contains helper methods, e.g., conversion Bitmap image format into a format Mat OpenCV and back and Mat download from a resource identifier for the resource.

Since we are working with a limited amount of memory, it is advisable to load an image with a lower resolution into the memory. The version with the reduced resolution should correspond to the size of the component of the user interface that displays it. A high-resolution image takes up a large amount of memory and does not provide a visible advantage when displayed. The choice of the option can be made by the user taking into account the characteristics of a particular mobile device. Figure 2 shows the program's pseudo code, which implements the algorithm described in Section 2, written with regard to the features of programming on mobile devices.

To work with bitmaps, we use the type Bitmap, which allows you to maintain the responsiveness of the user interface and reduce the amount of memory [12,13]. Next, for image processing, the Bitmap type is converted to the Mat type. Mat is a class for storing images that can be interpreted in C ++.

Using the BitmapFactory class, you can learn the resolution and type of graphics data before creating a Bitmap object and before allocating memory to that object. When you receive information about the resolution and size of the image, you can decide whether to load into memory the full-sized version of the image or its reduced version. We list some factors that need to be taken into account.

```
Start
for w ∈ (w_In, w_End)/c ∈ (c_In, c_End)/α ∈ (α _In, α_End) with Step_w/Step_c/Step_ α  do
start function1
                for i ∈ (1, N) with step = 1 do
                       for j ∈ (1, N) with step = 1 do
                $r = \left|\sqrt{\left(i-\frac{N+1}{2}\right)^2+\left(j-\frac{N+1}{2}\right)^2}\right|$
if  r > 0,5 then
$h(i,j) = \frac{\exp(-c\alpha\omega_1)}{\pi}\left(\frac{\sin\omega_1 r}{r}+\sin\omega_1 r+\frac{2\cos\omega_1 r}{r^2\omega_1}-\frac{2\sin\omega_1 r}{r^3{\omega_1}^2}+\left(\frac{\sin\alpha\omega_1 r-\sin\omega_1 r}{r}\right)+\left(\frac{c\cos\alpha\omega_1 r-r\sin\alpha\omega_1 r}{c^2+r^2}\right)\right)$ end
                         end
                end
           end
start function2
                   for i ∈ (1, N) with step = 1 do
                       for j ∈ (1, N) with step = 1 do
                       $XR(k,l) = XR(k,l) + X\left(k-\frac{N+1}{2}+i, l-\frac{N+1}{2}+j\right)h(i,j)$
                              end
                     end
          end
Input: min value of PSNR
New value of PSNR = psnr(XR, X, 1)
   if  calculated value of PSNR < previous value of PSNR and
   dispersion of processed image(XR) > dispersion of distorted image (X) then
    keep values of w/c/α
    keep value of PSNR
   end
Call function1 и function2 with new parameters w, c, α
     Output: Processed image.
End
```

Figure 2. The pseudocode of the program.

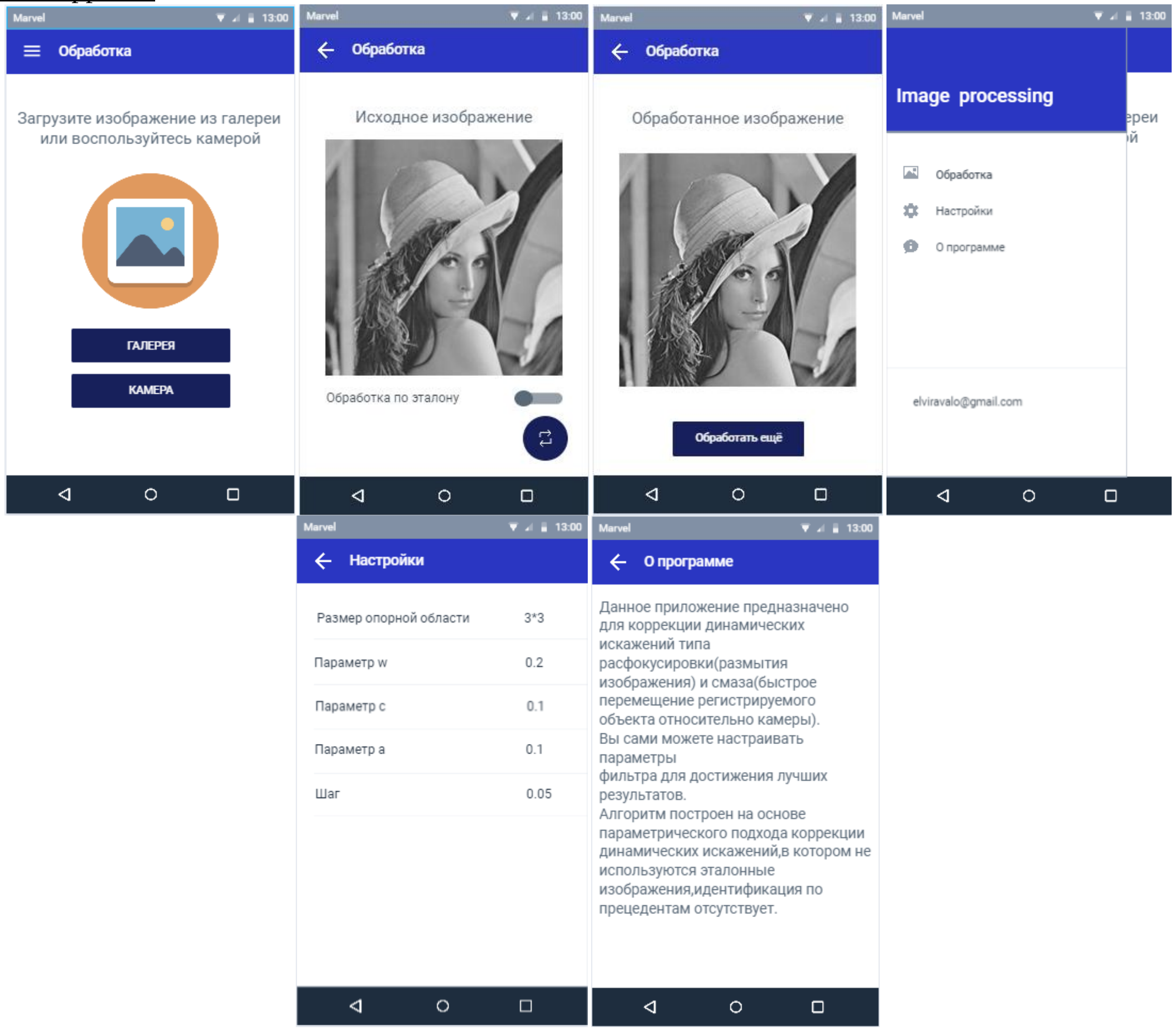
1. Estimated memory consumption when downloading the full version.
2. The amount of memory that can be spent on an image, considering the total memory consumption of the application.
3. Resolution of the component that will display the downloaded data.
4. The size and density of the screen points on the current device.

Methods of the BitmapFactory class should not be executed in the main thread of the user interface, so as not to reduce system performance. It is impossible to predict how long it will take to download data and processing them, it depends on various factors (speed of reading from the disk, the size of pattern to be displayed, processing power, and e.g.). If such a task will block the main UI thread, the system will mark the application as hovering, and you can get out of it, and even if removed.

If you want to perform operations that take some time, they must necessarily be performed in separate threads, called "background" or "worker" threads. To implement this task, we use the class AsyncTask, which offers a simple and convenient mechanism for moving laborious operations into the background thread.

The method of the AsyncTask class allows you to perform asynchronous operation in the user interface. About nor can perform minor image loading operations, file operations, database operations etc. in the workflow, and then publish the results in the main thread of the user interface without having to independently process the threads and/or handlers.

Figure 3 shows an application mockup built using an online tool for prototyping applications marvelapp.com.



Размер опорной области	3*3
Параметр w	0.2
Параметр c	0.1
Параметр a	0.1
Шаг	0.05

Данное приложение предназначено для коррекции динамических искажений типа расфокусировки(размытия изображения) и смаза(быстрое перемещение регистрируемого объекта относительно камеры). Вы сами можете настраивать параметры фильтра для достижения лучших результатов. Алгоритм построен на основе параметрического подхода коррекции динамических искажений,в котором не используются эталонные изображения,идентификация по прецедентам отсутствует.

Figure 3. Applicaion layers.

When you enter the application, the user enters the main screen, where he can choose how to load the image: choose the finished one from the "gallery" or take a photo using the camera. And for newer versions of Android, starting at 6, you need to allow access to the application to the camera. In the next step, the user can see the image he uploaded in the user interface component, make a decision, process whether it is an image by the standard, and click on the button that will start the process. After processing, the user goes to the screen where he can see the result. The processed image is saved in a special folder of the application, accessible through the built-in "gallery" application.

To change filter parameters for better results during processing, the user can go to the "settings" section of the application main menu.

To get information about the application, the user needs to go to the "About the Program" section.

The diagram of the main application classes that implement the described functionality is shown in Figure 4.

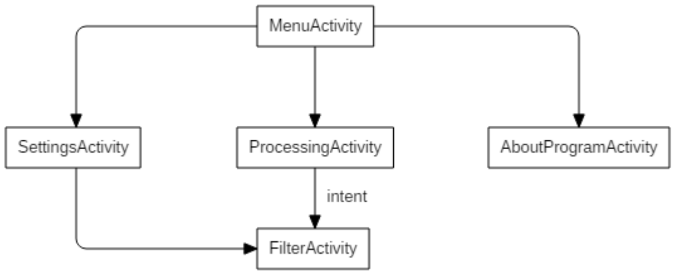


Figure 4. Class diagram.

4. Experimental results

Figure 4 shows the result of the filter with the configured parameters ω, c и a. Figure 6 (a) shows an example of a distorted halftone image by modeling a low-pass Gaussian filter with a blur: $\sigma = 3$. Figure 6 (b) shows the reconstructed image using a filter configured without a reference image, with a reference area dimension of 7×7. The result achieved during the restoration: PSNR = 25.13.

To quantify the achievable quality of recovery, we used test images (undistorted and distorted). We emphasize that the original undistorted image was not used to adjust the filter and restore it. We carried out a "blind" configuration.

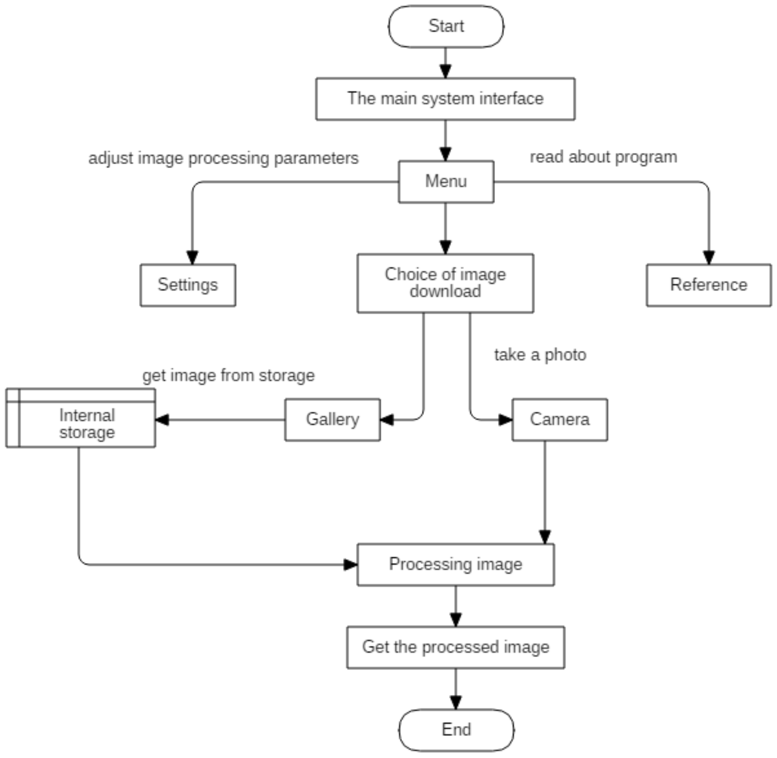


Figure 5. The scheme of the application.

Figure 5 shows a detailed scheme of the application.

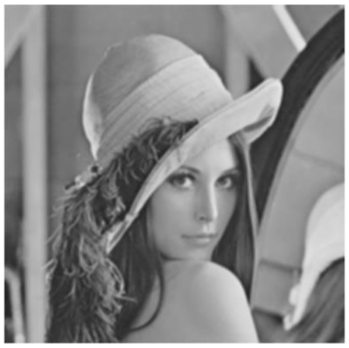
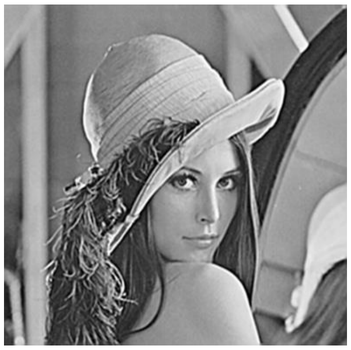
(a)  (b) 

Figure 6. Images of "Lena": (a) - distorted with a blur σ = 3, (b) – processed.

This algorithm for image restoration without using a reference image can also be used to restore color images. In this case, the processing is performed using the same algorithms for each component.

5. Conclusion

The method of correction of dynamic distortions is based on the "blind" identification of the parameters of the restoring filter, which does not require the reference image. A program is developed on the Android platform, designed for image processing in mobile devices. An example of processing test image obtained by modeling the lowpass Gaussian filter is given. The result is achieved when reconstructing the image with a blur: σ = 3 by PSNR = 25, 13. The results obtained may be of interest to users of tablets and smartphones. Further plans of the authors are in order to make it accessible to the wide range of users.

6. References

[1] Mail.Ru Group, Mobile Internet in Russia (Access mode: https://corp.mail.ru/media/files/40314-researchmob i l e mail.pdf)

[2] Greisukh G I, Ezhov E G, Kazin S V and Stepanov S A 2017 Diffractive elements for imaging optics of mobile communication devices *Computer Optics* **41(4)** 581-584 DOI: 10.18287/2412-6179-2017-41-4-581-584

[3] Greisukh G I, Ezhov E G, Kazin S V and Stepanov S A 2017 Single-layer kinoforms for cameras and video cameras of mobile communication devices *Computer Optics* **41(2)** 218-226 DOI: 10.18287/0134-2452-2017-41-2- 218-226

[4] Nikonorov A V, Petrov M V, Bibikov S A, Kutikova V V, Morozov A A and Kazanskiy N L 2017 Image restoration in diffractive optical systems using deep learning and deconvolution *Computer Optics* **41(6)** 875-887 DOI: 10.18287/2412-6179-2017-41-6-875-887

[5] Soifer V A 2010 *Computer Image Processing, Part II: Methods and algorithms* (VDM Verlag) p 584

[6] Pratt U 1982 Digital image processing (Moscow: Mir) **2** p 480

[7] Fursov V A and Yakimov P Y 2017 Internet technology for correcting dynamic distortions on images in mobile devices *Proceedings of the XIX All-Russian Scientific Conference* (Moscow: IPM them. M V Keldysh) 436-445 DOI: 10.20948/abrau-2017-09

[8] Fursov V A 2018 Construction of quadratic-exponential FIR filters with an extended average frequency response region *Computer Optics* **42(2)** 297-305 DOI: 10.18287 / 2412-6179-2018-42-2-297-305

[9] *Class AsyncTask* (Access mode: http://developer.alexanderklimov.ru/android/theory/AsyncTask .php)

[10] *Android developers* (Access mode: https: // developer.android.com/index.html)

[11] Paramonov I V 2013 *Development of mobile applications for the Android platform: a tutorial* (Yaroslavl: YarSU) p 88
[12] Hardy B, Phillips B, Stuart K and Marsicano C 2016 *Android. Programming for professionals* (St. Petersburg: Peter) p 640
[13] Howse J 2013 *Android Application Programming with O penCV* (Packt Publishing Ltd.) 1-131

Acknowledgements
The work was supported by the Ministry of Education and Science of the Russian Federation and the Russian Foundation for Basic Research, projects No. 17-29-03112, No. 16-07-00729-a.