NEOVISION
Industrial Vision Systems

MicroFocus

Version 2.0

web: http://www.neovision.cz, e-mail: microfocus@neovision.cz

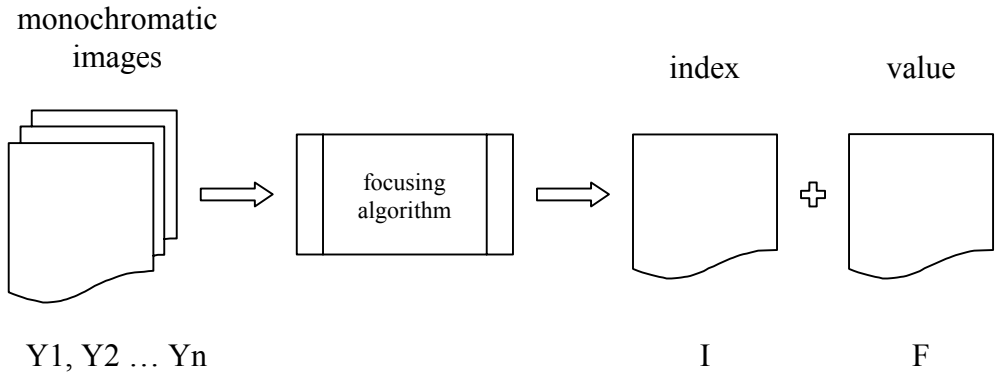
Table of contents

1. MicroFocus overview

1.1 The principle of focusing

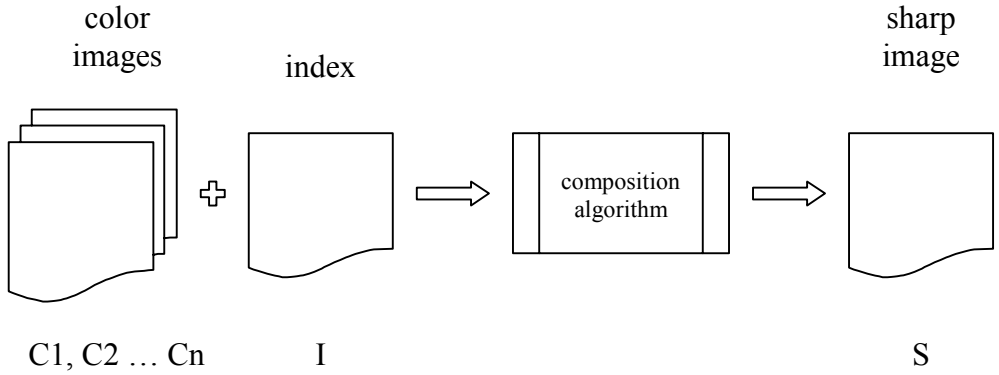
The focusing algorithm is more precisely an algorithm to increase the depth of field from a sequence of shallow depth of field images. Given a sequence of images ordered in the depth axis, the focusing algorithm computes the final sharp, large depth of field image.

Although the algorithms are designed to work for both grayscale and color images with various bits per pixel, internally the focusing takes only single monochromatic channel of image data. For each pixel coordinate in the result sharp image, the algorithm chooses the image from the sequence at which the pixels were most sharp. The indexes of such images for all image pixels are stored in so called "index image". To choose the best image index, a focusing feature value is computed for each pixel in each level of input sequence and the image with maximal feature value is selected. The feature values for all image pixels are stored in so called "feature image".



It is possible that some pixels are not sharp enough in any of input images – there are "holes" in index image. In that case, the index image contains index = -1 at such position. There are some algorithms available that can guess the correct index for some types of holes, i.e. close some holes.

Given the index image, one can easily construct the final, sharp color image by copying the pixel colors from appropriate input images according to the index value stored at given position.



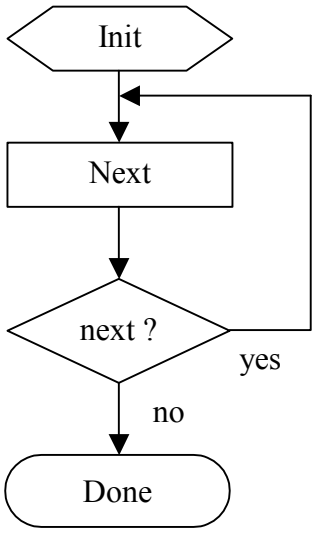
Where sharp pixel at [x,y] position is taken from I[x,y]-th color input image. Pixels at holes are best approximated by average pixel value at that position from all input images.

1.2 Incremental algorithms

When working with many large color images, the memory requirements must be taken into account. For example, if we want to focus from 50 true-color images at the resolution 1600x1200 pixels, we must process about 290 MB of pixel data! Although many of modern computers have such amount of memory available, increasing the size of memory required also slows down the algorithm as more and more data gets out of internal caches.

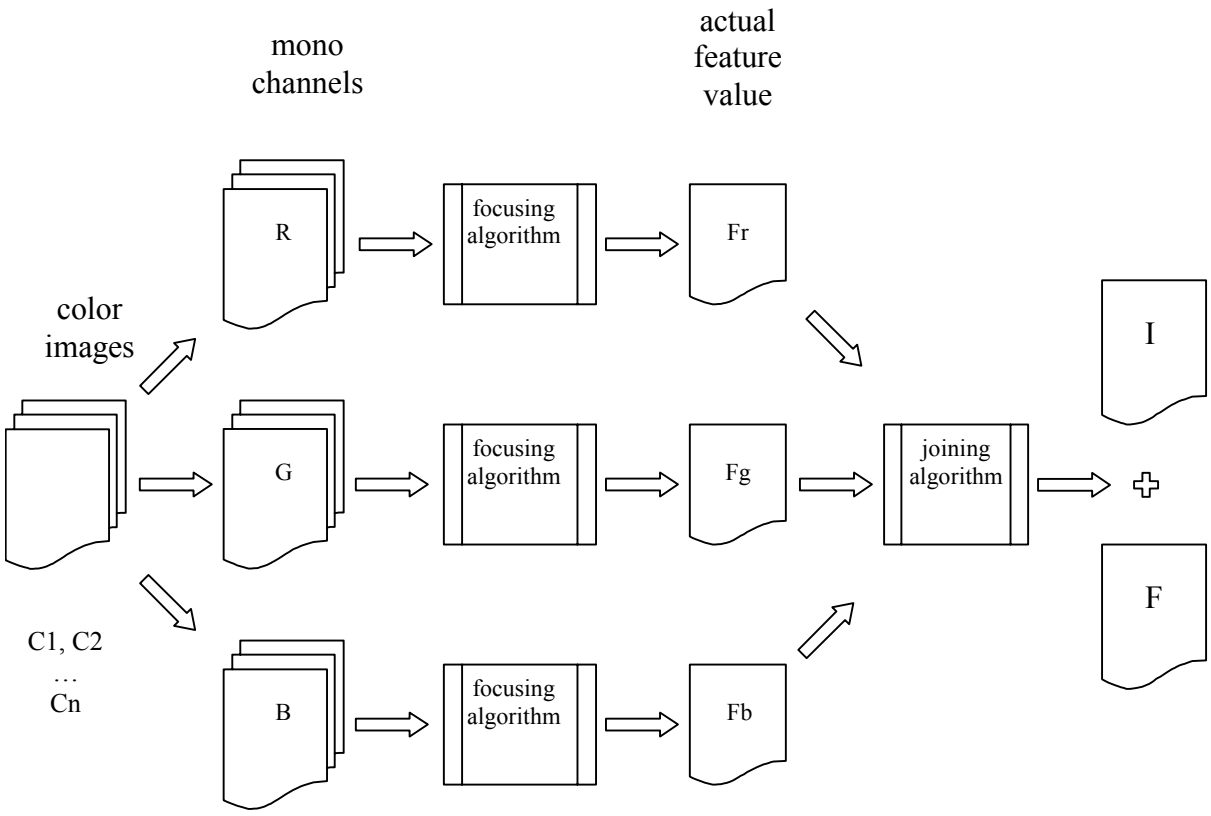
To reduce the memory requirements and speed up the processing, the both focusing and composition algorithms were designed to work as incremental ones. Incremental algorithms can construct its result in steps – they do not require all input images to be present in memory during its work. With incremental algorithm, after initialization step you simply call the "focus next" or "compose next" function, which takes just next input image and updates its result. After each "next" call, the results are valid for all the images passed to the algorithm so far. This allows you to display intermediate results after each step of focusing, so the user can see the focusing progress when more and more images are inserted. Incremental algorithms have memory requirements fixed, independent on number of input images.

Instead of loading all input images into memory and then calling single function which will focus them, using the incremental algorithm we can initialize the focusing task and then load in a loop next image and pass it to the focusing algorithm. When all images were passed, the result is final sharp image.



1.3 Singlechannel and multichannel focusing

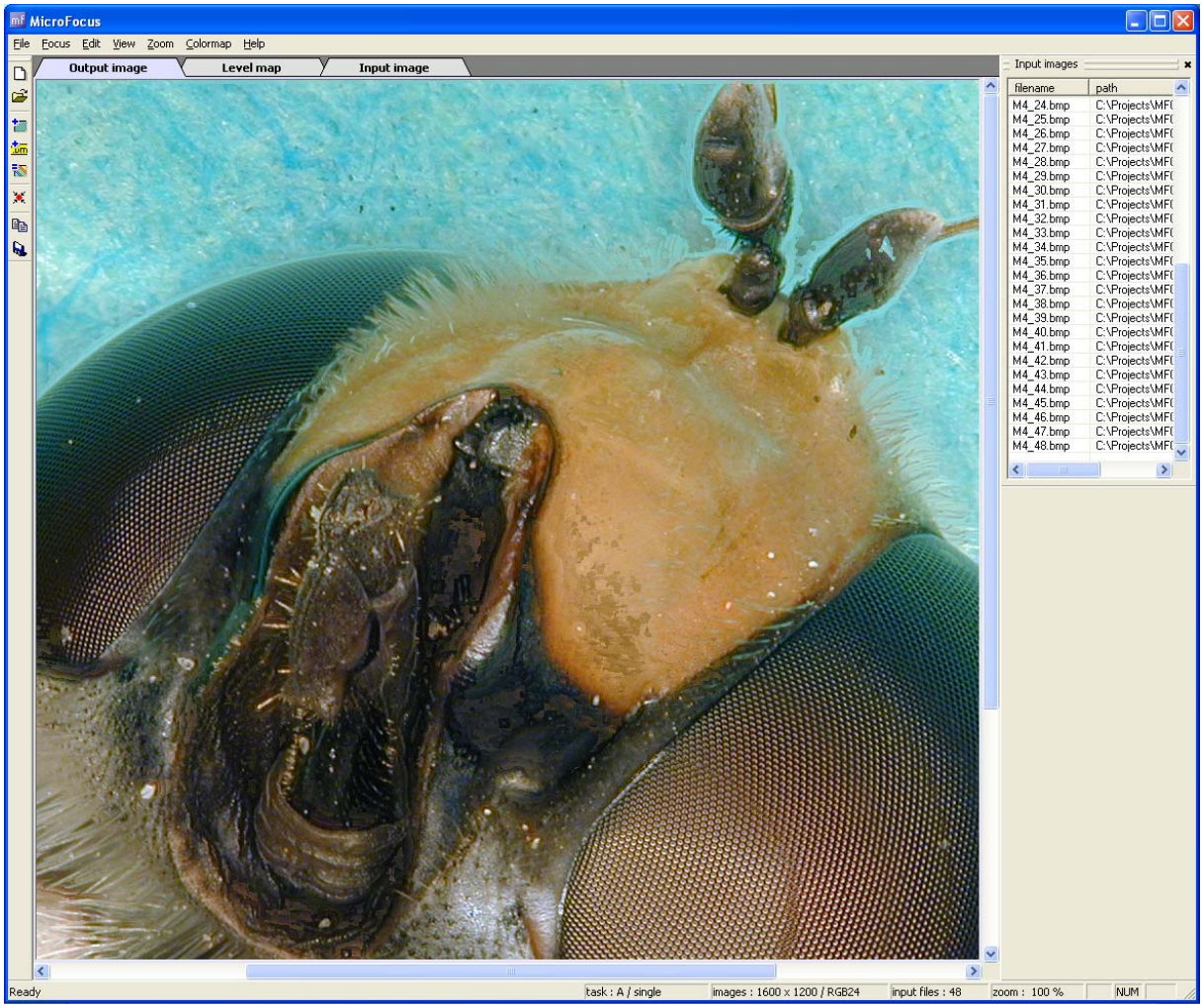
Color images can focused basically by two ways, called singlechannel and multichannel focusing. When using singlechannel focusing, the color images are first converted to gray images and focused. Resulting index image is used to compose sharp image, but as inputs are taken original color images, so final sharp image is again color. Multichannel focusing takes other approach. Color images are split to separate red, green and blue channels and on every channel is run a separate focusing task. Computed feature values in every channel are then joined to final feature value and index image.



Multichannel focusing task has no index output, indices are determined by joining algorithm. Outputs of joining algorithm are similar to outputs of normal (singlechannel) focusing task, inputs are feature values in every step of incremental focusing in separate channels. There are available two basic methods how to join channels – sum the channels and select maximum from channels.

2. User's guide

Main MicroFocus window is composed from several parts. There is a menu, toolbar, status bar, list of input images and image displaying area. The toolbar and input images list can be docked to any edge of main window or can be floating anywhere on your screen. You can customize the layout of main window, its position and size as you like it. Your settings are stored in registry and when you start the application again, it will be restored.



2.1 Description of menu commands

File submenu	
Open task	Open prepared task file and execute it.
Save image	Save image on currently selected tab to a disk.
Save level map	Save level map to a .mat file (Matlab format)
Exit	Quit the application
Focus submenu	
New focusing task	Create new focusing task.
Add image(s)	Add images to focusing task. User directly selects images.
Add image(s) specified by mask	Add images to focusing task. A working folder and file mask specify images.
Watch folder for image(s)	Watch specified folder for a new files, and insert them automatically to focusing task.
Finish task	Finish focusing. Result image is finalized, no more input images can be added.
Close gaps and finish task	Try to fill holes in index image and then finish focusing. Result image is finalized, no more input images can be added.
Edit submenu	
Copy	Copy image on currently selected tab to clipboard.
View submenu	
List of input images	Toggle visibility of input images list.
Toolbar	Toggle toolbar visibility.
Status Bar	Toggle status bar visibility.
Zoom submenu	
25 % - 400 %	Select a new zoom for displaying image.
Fit to window	Set zoom to fit entire image into window.
Colormap submenu	
Gray ... Flag	Choose a colormap for index image.
Help submenu	
About MicroFocus	Display information about program version.

2.2 Image tabs

The area for displaying images has three tabs, on each tab is displayed different image.

Output image	Displays a result of focusing. Result is valid for all the input images added to focusing so far.
Level map	Displays an index image. The color of every pixel is derived from its input image index.
Input image	Displays plain input image. When focusing it is always the last input image loaded. When you select an input image in a list, this image is loaded and automatically displayed in this tab.

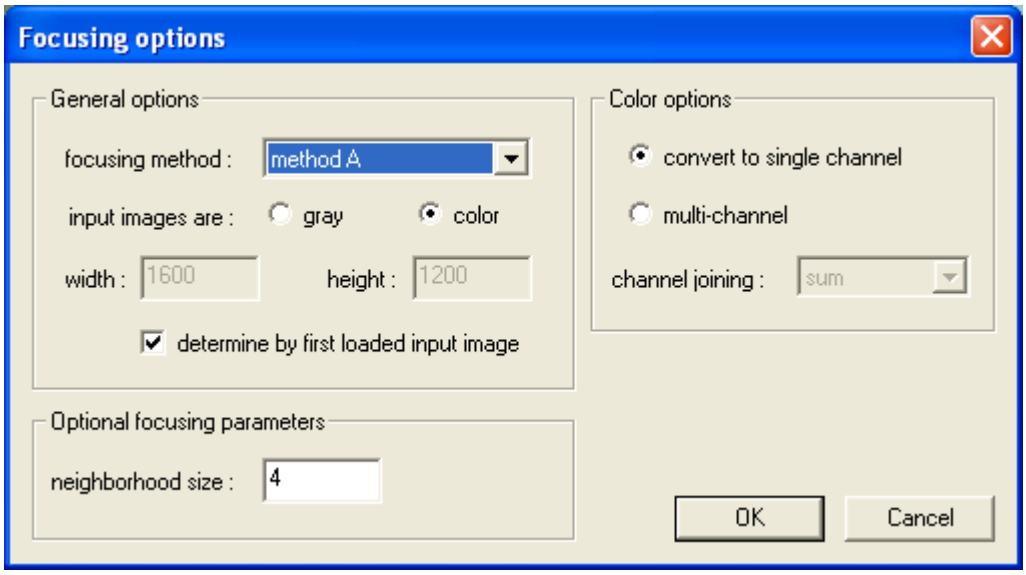
When the image is larger than are for displaying it, the scrollbars are active and you can scroll the window contents. Actual zoom can be changed by a menu commands.

2.3 Description of toolbar icons

	New focusing task.
	Open existing focusing task file.
	Add image(s).
	Add image(s) specified by mask.
	Finish focusing.
	Close gaps and finish focusing.
	Copy image to clipboard.
	Save image to disk.

3. New focusing task

When you start a new focusing, you can choose focusing options.



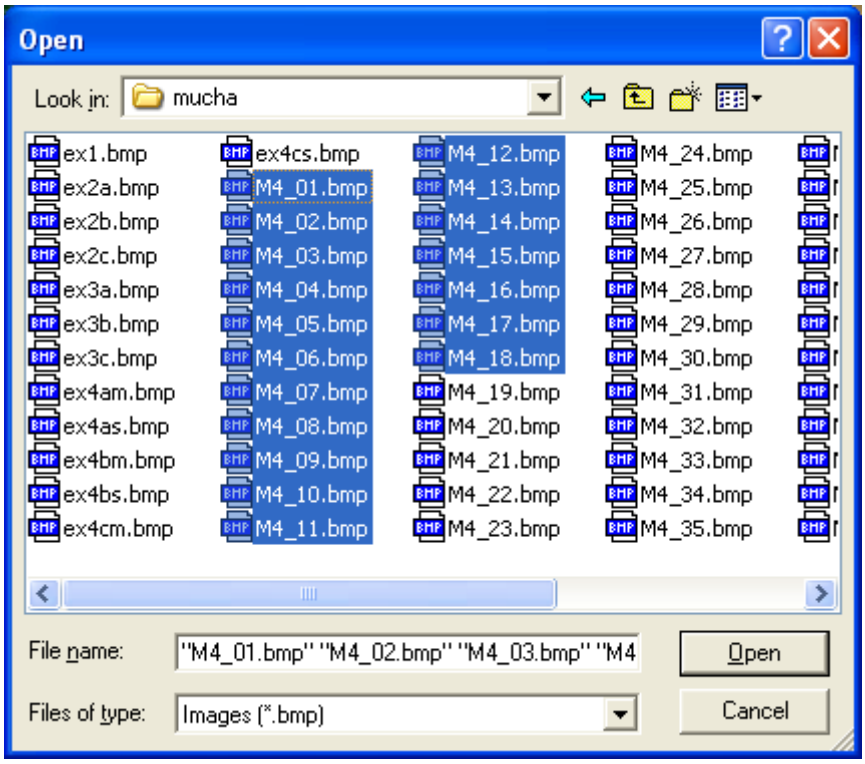
focusing method	You can choose between focusing methods A, B or C. For description of these methods, see chapter “Description of focusing methods”
input images are	You can specify type of input images – gray or color images.
width, height	You can specify width and height of input images, or let the program to determine dimensions from first image added to focusing task. When this checkbox is checked, the initialization of focusing task is deferred until first image is inserted to task.
neighborhood size	Neighborhood size for focusing method A, for other methods the value is ignored.
convert to single channel	Option to work in singlechannel mode. When input images are color, they are converted to gray and then focused.
multi-channel	Option to work in multichannel mode. Input images are split to separate channels, every channel is focused separately and then joined together to produce single index image.
channel joining	You can choose between two methods of channel joining. Sum of channels and maxima of channels is available.

4. Add images to focusing task

Images can be added by a three ways. You can select them directly, specify folder and mask and watch any folder for new images and add them automatically.

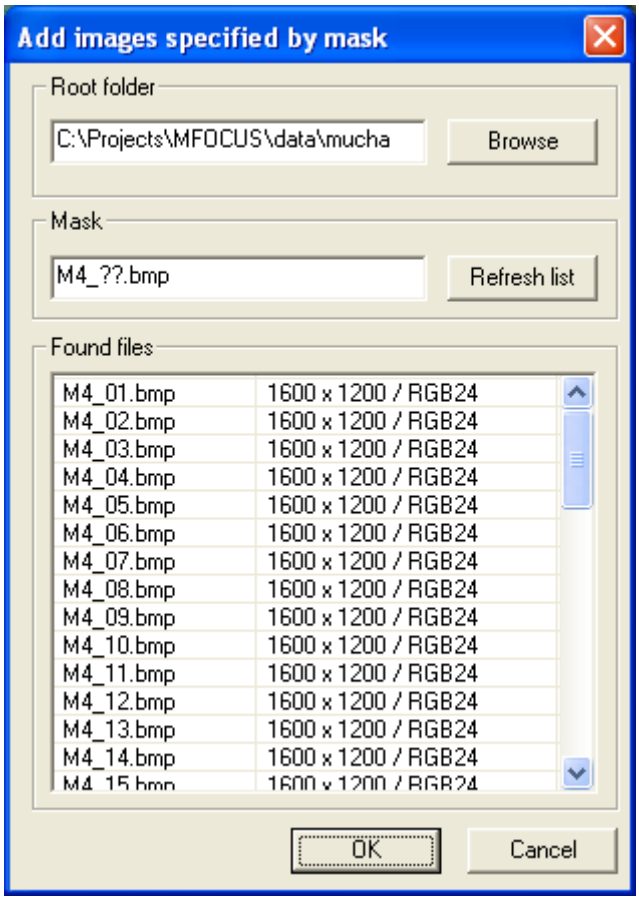
4.1 Add image(s)

For a direct selection of input images, the common Open File dialog is used. You can navigate to folder where your images are stored and select one or more images you wish to add to focusing task. Images are added to task in the order as they are placed on “File name:” edit line in the dialog.



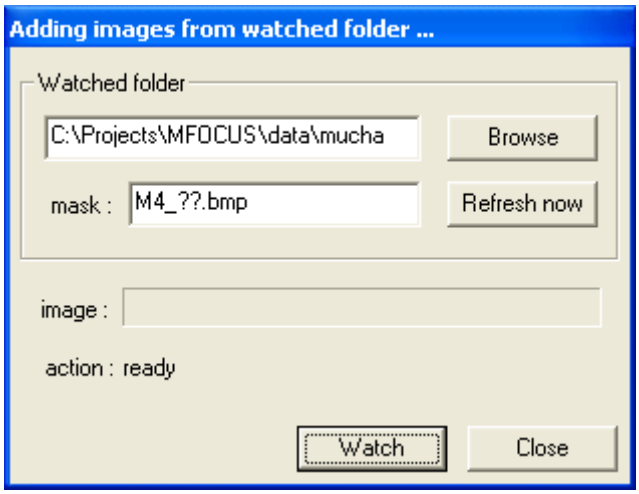
4.2 Add image(s) specified by mask

Other way how to add images is to specify a folder where they are stored and filename mask.



4.3 Watch folder for new images

Specify watched folder, filename mask and press Watch button. New images in the folder are automatically added to focusing task.



5. Finish focusing

If all input images were added, the focusing task should be finalized. This function will add remaining images held in memory into composition task. After usage of this function, no more images can be added to focusing.

6. Close gaps

Holes in index image can be closed (filled) in some circumstances by indices of its neighbors. When the holes are closed, entire composition task must be started again from scratch and it requires loading all input images again from disk.

7. Focusing task automation

Focusing can be automatized if you prepare a focusing task file. It is a common text file which specifies focusing options, input images and output file name. Focusing task file can be run by "Open task file" command or if you use its name as argument on command line, the task is opened and run.

section [Options]	
method	focusing method, can be one of 'A' for method A or 'B' for method B or 'C' for method C
type	input images type, can be one of 'RGB24' for color images or 'Y8' for gray images
width	width of input images, or 0. When set to 0, the image dimensions (width and height) are determined by first image added to task.
height	height of input images. When set to 0, the image dimensions (width and height) are determined by first image added to task.
neighborhood	neighborhood size for method A, can be between 1 and 50
mode	can be one of 'single' for singlechannel focusing or 'multi' for multichannel focusing
join	joining method for multichannel focusing, can be one of 'sum' for sum joining or 'max' for maxima joining
section [Files]	
folder	folder where input files are stored
mask	filename mask of input images
section [Output]	
close gaps	can be of 'yes' – function "Close gaps" will be used or 'no' – function "Close gaps" will not be used
save	filename of resulting sharp image
quit	can be one of 'yes' – after task execution the app will be closed or 'no' – the app will not close and stays running

Example of focusing task file:

```
; Microfocus task file
; Technology for processing shallow depth of field image data
; using Mfocus library (c) 1998 - 2002, Neovision s.r.o., Prague, CR

[Options]
method       = A
type         = RGB24
width        = 0
height       = 0
neighborhood = 2
mode         = single
join         = sum

[Files]
folder       = c:\projects\mfocus\data\mucha
mask         = M4_??.bmp

[Output]
close gaps   = yes
save         = mucha.bmp
quit         = no
```

8. Description of focusing methods

The methods differ in how the feature value is calculated. Basically, the feature value is dependent on some neighborhood of evaluated pixel. The neighborhood may be 2D only or 3D - the feature is calculated not only from current image but also from previous and following images in the sequence. If the feature value is dependent on 6 previous and 6 following images, the 6 first and last 6 images in input sequence cannot get a feature. Those images are considered as margins only, their pixels will never be in final image. The best results can be achieved when the marginal images do not contain any focused data, ie. the input images sequence starts with totally unfocused image and ends up with unfocused image.

8.1 MF_METHOD_A

Method properties:

parameters	n = neighborhood size in x and y (int n = (int) params)
neighborhood in x and y	n (2n+1 x 2n+1 window)
neighborhood in depth	no
memory requirements	no other internal memory allocated
speed	fast

Remarks

Method “A” works only on 2D neighborhood of selectable size. Due to this, you can do focusing and composition in single pass. Input images needn’t to be equidistant, there are no marginal images. This means the first and last image in sequence may contain sharp pixels.

8.2 MF_METHOD_B

Method properties:

parameters	none
neighborhood in x and y	4 (9 x 9 window)
neighborhood in depth	6 previous and 6 following images
memory requirements	approx. 12 cached preprocessed images and 3 integer images
speed	medium

Remarks

- input images should be equidistant in depth
- first 6 and last 6 images are marginal, no pixels will be taken from them
- marginal images should not contain sharp pixels

8.3 MF_METHOD_C

Method properties:

parameters	none
neighborhood in x and y	4 (9 x 9 window)
neighborhood in depth	5 previous and 5 next images
memory requirements	approx. 12 cached preprocessed images
speed	slow

Remarks

- input images should be equidistant in depth
- first 5 and last 5 images are marginal, no pixels will be taken from them
- marginal images should not contain sharp pixels