

Help Contents

Image Arithmetic
version 2.1
by Richard van Paasen ✧

The Art of Huge Image Arithmetic

Image Arithmetic Copyright © 1997
by Richard van Paasen
All Rights Reserved

[Tutorial](#)

[Disclaimer and rules](#)

[Registration](#)

[Future releases](#)

[Troubleshooting](#)

[Bug reports](#)

[Acknowledgments](#)

Arithmetic Tutorial

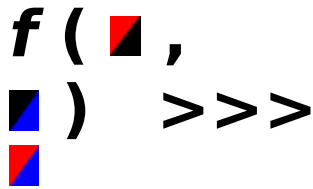
Image Arithmetic
version 2.1
by Richard van Paasen ✧

Merge
Tile
Quantize

Preferences
File Formats
Registration

Merge

Merges two images, channel by channel to create a new image:



To merge images, follow these steps:

Step 1: Select the merging method.

Eight methods are available:

<i>Method</i>	<i>Formula</i>	<i>Description</i>
Add	Image1 + Image2	This function adds colour values.
Subtract	Image1 - Image2	This function subtracts colour values of image 1 from colour values of image 2.
Multiply	Image1 x Image2	This function multiplies colour values.
Difference	absolute value of Image1 - Image2	This function computes the difference between colour values.
Lighter	MAX(Image1 , Image2)	This function selects the highest colour value of image 1 and image 2.
Darker	MIN(Image1 , Image2)	This function selects the lowest colour value of image 1 and image 2.
Horizontal interlace	column-by-column concatenation of image1 and image2	This function interlaces the source images horizontally.
Vertical interlace	row-by-row concatenation of image1 and image2	This function interlaces the source images vertically.

Step 2: Select source images.

Select two source images by pressing the Load Image buttons. Image Arithmetic then checks the image file format. If the format is correct, the image name, width, height and number of colours is displayed next to the button.

Step 3: Select the channels that will be merged

Select the channels (Red, Green and Blue and/or Alpha) of the source images that you want to merge. Image Arithmetic processes the selected channels from the source images and ignores the deselected channels. If the alpha channel is selected and one of the source images does not have an alpha channel, then a default value for the alpha channel will be used for that image. This default alpha can be set to either 0 or 255, depending on whether alpha should be interpreted as a mask or a transparency channel. The default alpha can be set in the preferences dialog.

Step 4: Set the weight factors, the divisor and the offset.

(Only for methods add, subtract, multiply and difference)

If the add, subtract, multiply or difference methods are selected, then weight values, a divisor and an offset must be set.

Weight Value

An integer factor by which the colour values of each image are weighted before applying the merge method.

Divisor

A floating point factor by which the result of the merge method is divided. The divisor can be calculated automatically by setting the *Lock divisor* option.

Offset

An integer factor that is added to the divided result of the merge method.

Colour values are calculated as depicted in the following formula:

$$\text{Result} = \left[\text{Offset} + \frac{(\text{Weight1} \times \text{Image1}) \text{ METHOD } (\text{Weight2} \times \text{Image2})}{\text{Divisor}} \right]$$

add, sub, mul or dif
↓
rounding
clipping

Thus, the merging method is applied in the following order:

- 1) first, the colour values of the source images are weighted,
- 2) then, the resulting values are added, subtracted, multiplied or differenced,
- 3) then, the resulting values are divided by the divisor,
- 4) then, if colour rounding is enabled, the resulting values are rounded,
- 5) then, if colour clipping is enabled, the resulting values are clipped.

Lock divisor

The divisor is calculated automatically if this option is set. Image Arithmetic sets the divisor to a value that produces colour values within the range [0 ... 255]. The value of the automatic divisor depends on the merge method and the weight values:

Method	Divisor
Add	weight1 + weight2
Subtract	max(1,weight1 - weight2)
Multiply	weight1 x weight2 x 255
Difference	max(1,weight1 - weight2)

Step 5: Select colour calculation options.

(Only for methods add, subtract, multiply and difference)

To control colour calculation, the following options can be selected:

Colour Clipping

Enabled: colour values that are larger than 255 are set to 255 and colour values that are smaller than 0 are set to 0.

Disabled: colour values that are larger than 255 are reduced by a multiple of 256 so that

the value ranges between 0 and 255 and colour values that are smaller than 0 are added a multiple of 256 so that the value ranges between 0 and 255.

The effect of clipping when colour values are <0 or >255 is depicted in the table below.

<u>Clippin</u> <u>g</u>	<u>Colour value</u> <u>< 0</u>	<u>Colour value ></u> <u>255</u>
Enabled	0	255
Disabled	$n \times 256 - \text{value}$	$\text{value} - n \times 256$

With colour clipping enabled, arithmetic can result in colour shifting (e.g. a value of 613 is clipped to 255. With colour clipping disabled, arithmetic can result in apparently random colours (e.g. a value of 613 becomes $513-256-256=101$).

Colour Rounding

Enabled: produces correct, rounded dividing results, thus rounding non-integer colour values up or down to the nearest integer.

Disabled: produces truncated dividing results, thus truncating non-integer colour values down to the nearest integer.

The effect of rounding is depicted in the table below for two example colour values.

<u>Roundin</u> <u>g</u>	<u>Colour value</u> <u>26.4</u>	<u>Colour value</u> <u>26.7</u>
Enabled	26	27
Disabled	26	26

Note that most graphics applications only provide the truncating method and thus do not produce arithmetically correct results.

Step 6: Select a name and a suitable format for the destination image

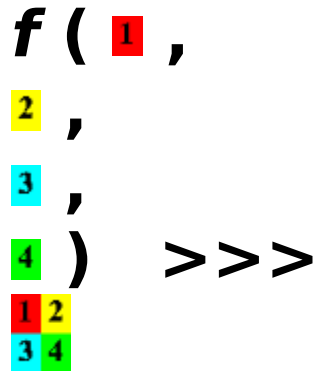
After pressing the Process button, you have to select a format and name for the destination image. The format is selected by specifying an appropriate extension in the filename. The following extensions are recognized:

<u>Format</u>	<u>Extension</u>
JPEG	.jpg, .jpeg
TARGA	.tga, .targa
PCX	.pcx
PNG	.png

Note that the JPEG and PCX formats cannot be used when you merge the alpha channel. This is because these formats don't support alpha channels.

Tile

Tiling arithmetic allows several images to be concatenated into one bigger image. In other words:



The source images must be numbered in a way so that Image Arithmetic knows how to layout the sub-images in the destination image. The naming convention of FractInt is used.

To tile images, follow these steps:

Step 1: Name your sub-images according to the FractInt naming convention

Each sub-image uses the file name *abcde_XY.ext*. The components of the filename are:

<u>Name part</u>	<u>Description</u>
abcde_	Any valid string of characters
X	the column (left to right)
Y	the row (top to bottom)
.ext	extension of the images, .png, .tga, .jpg or .pcx

Columns and rows are indexed by a single character. The indexing starts at 0 and goes through 9, and then if needed, the letters of the alphabet are used (i.e., 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, and z). The maximum size of the assembled image thus can be 36 x 36.

In the example that is given above, the naming of the sub-images is:

<u>Image #</u>	<u>Filename</u>
image 1:	image_00.png
image 2:	image_10.png
image 3:	image_01.png
image 4:	image_11.png

The number of columns and the number of rows do not necessarily need to be the same, of course, and can even be one. Note that all sub-images must be stored in the same folder.

Step 2: Select a source in Image Arithmetic

Press the Source button and select one of the sub-images. Image Arithmetic will now:

- check the sub-images' file format(s),
- check whether all sub-images are present,
- count the number of rows and columns in the final image,
- compute the width and height in pixels of the final image,
- check whether all sub-images use the same colourmap

Step 3: Check/uncheck the *write colour mapped image* option

If all sub-images that are to be assembled are colour mapped images and if they all use the same colour map, Image Arithmetic can generate a colour mapped destination image. If the colour mapped sub-images do not have the same colour map, or when at least one sub-image is a truecolour image, a truecolour destination image will be generated.

Image Arithmetic automatically detects whether all sub-images use the same colour map and (de)selects the colour map checkbox accordingly. This choice can be overridden by manually (de)selecting the colour map checkbox. However, a colour mapped image can not be generated if one or more sub-images is a truecolour image. Image Arithmetic does currently not support colour quantization.

Rarely, you may encounter that Image Arithmetic does not check the colour map checkbox while you are sure that all sub-images use the same colour map. This can happen when there is a sub-image that does not use all different colours of its colour map. The entry in the colour map that is not used may contain any colour, because it is not used after all. Image Arithmetic checks all entries of the colour maps of all sub-images. If it encounters a palette that has some entries with a different colour (maybe because they are not used by the corresponding image), Image Arithmetic deselects the colour map checkbox. You may select it manually if this occurs.

Step 4: Select a name and a suitable format for the destination image


After pressing the OK button, you have to select a format and name for the destination image. The format is selected by specifying an appropriate extension in the filename. The following extensions are recognized:

<u>Format</u>	<u>Extension</u>
JPEG	.jpg, .jpeg
TARGA	.tga, .targa
PCX	.pcx
PNG	.png

Note that the JPEG format can only be used when true colour images are written.

Quantize

The colour quantization function reduces the number of different colours in an image. True colour images can be converted to colour mapped images by this function. The quantization algorithm of Image Arithmetic applies statistical methods to determine the optimal colour map for the quantized image.

f (.) >>>


To reduce the number of colours in an image, follow these steps:

Step 1: Select a source image

Select a source image by pressing the Source buttons. Image Arithmetic checks the image file format and complains when it does not support it. If the format is correct, the image name, width, height and number of colours is displayed next to the button.

Step 2: Enter the desired number of colours

Any number between 2 and 256 can be chosen.

Step 3: Select a quantization method

Three methods are provided:

Standard quantization

This method generates a standard 3-3-2 palette that spans the entire RGB colour spectrum. Standard quantization does not generate an optimal palette, but it is a fast method and it doesn't need much memory.

Octree quantization

This method generates an optimal palette. Octree quantization doesn't need much memory. Octree quantization is described in: *A Simple Method for Color Quantization: Octree Quantization*, M. Gervautz and W. Purgathofer (*Proceedings CG International*, '88).

Median-Cut quantization

This method generates an optimal palette. Median-Cut quantization requires much memory. The amount of memory needed is determined by the precision at which the colour histogram is built. Median-Cut quantization is described in: *Color Image Quantization for Frame Buffer Display*, P. Heckbert (*SIGGRAPH 82*). Note that the Median Cut method can only quantize reliably if the image does not contain colours that appear more than 65535 times!

Step 4: Set additional options

Depending on the selected quantization method, the following options may be selected:

Include Windows' colours

If the quantized image is to be used in the Windows environment, you may want to include the Windows standard palette in the palette of the image. This option can not be used with the Standard quantization method

Floyd-Steinberg dithering

After calculating a palette for the image, the image's colours will be mapped to that palette. The error in the mapping (the difference between the original colour and the palette colour) can be propagated to neighboring pixels according to the Floyd-Steinberg method. This method reduces colour banding.

Reduce colour bleeding

With some images (e.g. computer generated 3D graphics), Floyd-Steinberg dithering produces bleeding colours at sharp edges with high contrast. This option will reduce colour bleeding by monitoring and reducing the propagated error terms of the Floyd-Steinberg method.

Step 5: Set the Median-Cut precision

If the Median-Cut quantization method is selected, then the precision of the histogram (per colour channel) must be given. The precision per channel is counted in bits and must range between 1 and 8. Higher precision means better results, but also requires (exponential) more memory and computing power. The amount of memory that is needed for different settings of precision is listed in the table below.

<i><u>Precisio n</u></i>	<i><u>Amount of memory</u></i>
5-5-5	64 KB
5-6-5	128 KB
6-6-6	512 KB
6-7-6	1 MB
7-7-7	4 MB
7-8-7	8 MB
8-8-8	32 MB (!)

Satisfactory results can almost always be obtained with precision 5-6-5 or 6-6-6.

Step 6: Select a name and a suitable format for the destination image

After pressing the OK button, you have to select a format and name for the destination image. The format is selected by specifying an appropriate extension in the filename. The following extensions are recognized:

<i><u>Format</u></i>	<i><u>Extension</u></i>
TARGA	.tga, .targa
PCX	.pcx
PNG	.png

Note that the JPEG format cannot be used, because it doesn't support colour mapped images.

Preferences

The following preferences/options can currently be selected:

Remember open/save folders

If this option is selected, the folders where images were retrieved and stored will be remembered by Image Arithmetic next time you start the application.

Remember arithmetic options

If this option is selected, the options that were set throughout the application will be remembered by Image Arithmetic next time you start the application.

Progression indication

Image Arithmetic shows processing time and progression percentage while it processes images.

Select *per completed scanline* if you want this information after each processed scanline.

Select *per 5% progression* if you want this information per 5% of the total number of scanlines to process.

Default alpha channel

Some Image Arithmetic functions can use alpha channels in addition to red, green and blue channels. However, not all image formats (e.g. PCX and JPEG) support alpha channels. A default value alpha value must be defined for these formats so that they can be used in conjunction with images that support alpha channels. The possible values for default alpha are currently 0 and 255 (minimum and maximum alpha values). Please note that the interpretation of the alpha channel is not standardized. The alpha channel is interpreted as mask channel or as transparency channel. Refer to the manual of your graphics software to check which interpretation is used.

File Formats

Supported file formats

Image Arithmetic supports four widely used image formats. These formats can be read and written:

Format	Comment
JPEG	All JFIF compliant formats
PCX	All versions
TARGA	All non-upside down versions
PNG	PNG 8, 16, 24, 32, non-interlaced formats.

Suitable file formats

Image Arithmetic lets you select a file format for destination files by specifying the proper file extension. The colour capabilities of the file formats are listed in the table below.

Format	Extension	Format is suitable for storing		
		Colour Mapped	True Colour	True Colour + alpha
JPEG	.jpg, .jpeg	<i>no</i>	YES	<i>no</i>
TARGA	.tga, .targa	YES	YES	YES
PCX	.pcx	YES	YES	<i>no</i>
PNG	.png	YES	YES	YES

File format options

Some file formats require additional settings. These can be set in the *File formats* dialog:

TARGA compression

Select the *compressed TARGA* option to turn on RLE-compression for writing TARGA files.

TARGA truecolour is 16 bits

Select the *truecolour is 16 bits* option to always write 16 bits truecolour TARGA files instead of 24/32 bits TARGA files. Be careful when selecting this option because Image Arithmetic will convert 24/32 bit colours to 16 bit colours without dithering. Only select this option if you really need to, for example, save harddisk space.

JPEG DCT accuracy

The *accurate DCT* option controls the JPEG de/encoding accuracy. Select this option to produce better JPEG results, at cost of slower execution.

JPEG compression quality

The *compression quality* for JPEG compression can be set. A higher quality implies less compression and thus larger files. The range of the compression quality is 1...99. A reasonable quality is 75. Note that the image loses detail when the compression quality is too low.

PNG compression level

The *compression level* for PNG compression can be set. A higher level implies higher compression ratio and thus smaller files. However, higher compression also implies slower compression. The range of the compression level is 0...9. A reasonable factor is 6. The quality of the image is not affected by the compression level.

PCX horizontal and vertical DPI

PCX files store the *horizontal and vertical DPI* at which they were scanned or generated. These values default to 300.

Additional notes

Image Arithmetic does not support interlaced images

Image Arithmetic's purpose is to be able to handle very high resolution image files (i.e., "BIG") with minimal memory requirements and robust processing. Such big files *will never be used* for, for example, display on Web pages or for streaming display. The big files *will be used*, though, to make some sort of print or transparency on an output device where interlaced files either cannot be used or they confer no advantage. Of course, a small number of users might rarely have a need to tile a bunch of small images, each of which is interlaced. In this case, you could use "Graphic Workshop" or some other batch processor to first convert the interlaced images to non-interlaced.

Interlaced images are not sequentially stored. To de-interlace an image, a copy must be made in memory, or on disk. Because Image Arithmetic targets arithmetic on large images, this may cause problems with the amount of (virtual) memory that is needed to de-interlace images. Be sure to write your source images non-interlaced. If you have interlaced images, de-interlace them first!

Image Arithmetic does not support upside-down or leftside-right images

For the same reason as Image Arithmetic does not support interlaced images. Upside down, or leftside-right stored images must be flipped vertically or mirrored horizontally before they can be combined with non-upside down images. This requires a lot of disk space and RAM memory. Use a program, like Alchemy Mindworks Graphics Workshop, to convert or create images that are stored in the proper orientation. Note: Paint Shop Pro creates upside down TARGA images.

Which image format and compression to choose

Which format you should choose for storing images depends on your needs. Be careful when choosing a graphics format:

PCX

Compresses the image data poorly. It uses the RLE compression scheme, which performs best when the image contains large areas with the same colour.

TARGA

Compresses the image data poorly. TARGA uses a combined RLE/RAW compression scheme, which achieves approximately equal results as the RLE compression scheme. Furthermore, TARGA can store alpha channels.

GIF

GIF support is currently disabled because of Unisys' patent on the LZW algorithm. Please use the PNG format. PNG replaces GIF gracefully and it even compresses better.

PNG

Compresses the image data good. The compression/speed ratio of PNG can be configured without loss of image quality. Furthermore, PNG can store alpha channels.

JPEG

Compresses the image data excellent. However, this is done by *reducing the quality* of the image. Unless you have very little hard disk storage or unless you are creating graphics for Web pages, you shouldn't choose JPEG for storing images.

Troubleshooting

Q1: Image Arithmetic cannot read my images

Refer to the [file formats section](#) for a list of supported file formats. Some features, such as interlacing and upside-down images are not supported.

Q2: Image Arithmetic refuses to read GIF files

Due to Unisys' patent on the LZW algorithm, GIF support is disabled in Image Arithmetic. Please use the PNG format, which replaces GIF gracefully.

Q3: How can I view large images

Some graphics programs appear buggy when they have to display large images. Lview Pro and Paint Shop Pro, for example, work fine for smaller images, but they currently can't handle large ones. While testing Image Arithmetic, Alchemy Mindworks Graphic Workshop and Adobe PhotoShop were able to handle large images (other programs have not been tested by me: email me if you know programs that can display large images!). However, Adobe Photoshop may have problems with large compressed TARGA24 files.

Q4: My favorite application cannot read images that are created by Image Arithmetic

Image Arithmetic targets very large images, up to a resolution of 65535 x 65535. Some programs try to load images with such large resolutions into memory. Because these resolutions are definitely not standard, some applications are not designed to read these huge images. For example, Adobe Photoshop cannot display large compressed TARGA images. A work-around for this problem is described in the [file formats section](#).

Q5: Why is 65535 x 65535 the maximum resolution?

Because the image formats store the width and height of the image as integers of 2 bytes long. The maximum value that can be stored in 2 bytes is 2 to the power of 16, or: 65535. This restriction is explicitly defined in the image format specifications.

Q6: My computer locks up

That's an interesting problem. Please try to write down the conditions under which this occurs and send me a [bug-report](#).

Q7: More questions

If you have more questions, don't hesitate to contact me!

DISCLAIMER and REGISTRATION RULES

Please maximize the help window and read the following agreement carefully!

AGREEMENT

This agreement is applicable to both the unregistered and the registered version of Image Arithmetic. Use or possession of Image Arithmetic constitutes acceptance of this agreement. If you do not agree with the conditions and rules stated in this text, please remove your copy of Image Arithmetic.

author

Image Arithmetic is copyrighted to me: Richard L. F. van Paasen.
Refer to the author for my email address and WWW-homepage location.

responsibility

I am not responsible for any damage, dysfunction, loss of data, or any other unwanted effects that result from using Image Arithmetic, nor give I warranties of any kind, expressed or implied, in regard to Image Arithmetic. I will provide no remedy for indirect, consequential, punitive or incidental damages arising from the use or installation of Image Arithmetic.

evaluation period

According to the shareware-concept, you may evaluate Image Arithmetic for a period of 30 days. After this evaluation-period, you must (a) pay the registration-fee, or (b) remove the software from your computer.

reversed-engineering, modifications and copying

No component of Image Arithmetic may be reversed-engineered, modified nor copied. This applies to the distribution archive, the program itself, the example files, documentation files and whatever may be shipped in the distribution archive.

distribution - unregistered version

You may freely distribute the unregistered version of Image Arithmetic, but only in its original and archived form. You may, however, not sell the unregistered version of Image Arithmetic. Payment for the distribution medium (such as a CD-ROM) on which an unregistered version of Image Arithmetic is stored may not exceed US\$25.

distribution - registered version

You may not sell and/or distribute registered versions of Image Arithmetic without written permission of the author. Nor may you make the registered version of Image Arithmetic available to other people by any means.

distribution - registration keys

Registration keys may only be created and distributed by the author of Image Arithmetic.

registration

When you decide to register, it is assumed that you have tested the software to meet your requirements. Once a registration procedure has been fulfilled, it can not be undone. Any accepted payment can not be refunded.

Registration

If you like Image Arithmetic and/or continue to use it, you'll have to register! Registering shareware is a normal procedure and a moral obligation!

I've put a lot of time in developing, testing and debugging Image Arithmetic. If you continue to use Image Arithmetic, you'll save a lot of time and you can do things you otherwise couldn't do easily. Please support the development of Image Arithmetic by registering! In return, the registration reminders will not be displayed anymore and the application will show your name instead of "Unregistered Version".

By registering, it is assumed that you have read the disclaimer and that you agree with it!

Registration fee

A registration key is available for a small registration fee. You can choose from the following currencies:

Currency	Price per copy	Additional price for delivery on 1.44 disk
US dollars	US\$ 20	US\$ 5
Dutch Guilders	HFL 40	HFL 10
German Marks	DM 40	DM 10

The listed registration fees apply to single copies. If you need more than five registration keys, you may contact me for a reduced price. The current method of payment is by sending paper money by regular mail. Unfortunately, I can currently not accept cheques, credit cards or bank transfers. For small amounts like this, international bank transfers cost too much (about the same amount as the registration fee) Furthermore, banks in the Netherlands also charge me approximately \$15 for converting international cheques into cash.

If you insist on sending a cheque, or sending a bank transfer, I will accept it, but the registration fee will be higher then (contact me first!).

Registration procedure

To register Image Arithmetic, follow these steps:

Step 1: Fill in the registration form

Choose a name by which you want to register. This can be your name, your organization's name or something else. The registration name should have a minimum of 8 characters and a maximum of 32 characters. Write down your registration name and fill in the registration form. You may either print the registration form and send it by regular mail, or edit it with a wordprocessor and send it by email.

Step 2: Send the registration form and pay the registration fee

Put the registration fee in an envelope (wrap it in paper) and send it to the address below. Include a printed and filled-in copy of the registration form if you did not already email it.

address:

**Richard van Paasen
Moreelselaan 42
5643 RN Eindhoven
The Netherlands (Europe)**

email address:
rvpaasen@dds.nl

Step 3: Receive a registration key

As soon as I have received the payment, I will email you a registration key. If you cannot receive email nor access the World Wide Web, add US \$5, HFL 10 or DM 10 to the registration fee, and I will send you a copy of Image Arithmetic on 1.44 MB disk(s) by regular mail. This disk will then contain the latest version of Image Arithmetic and a registration key.

Step 4: Complete the registration

As soon as you received the registration key, you can turn the unregistered version of Image Arithmetic into a registered version. To do so, go to the registration dialog, enter your registration name in the *name* field, and the registration key in the *key* field. Image Arithmetic will then be registered immediately.

Registration form Image Arithmetic

Please fill in the information below. You may either print this form and send it by regular mail, or edit it with a wordprocessor and send it by email.

ORDERING INFORMATION

Number of copies: _____ (US\$ 20, DM 40 or HFL 40, per copy)

Want floppy disk: (yes/no) (US\$ 5, DM 10 or HFL 10, per order)

Total price: _____

Registration names: _____

Enter one name for each copy. Contact me for a reduced price if you need more than 5 copies.

CUSTOMER INFORMATION

Your name: _____

Organization: _____

E-Mail Address: _____

Address: _____

City, State, Zip: _____

Country: _____

Phone(s): _____

Where did you hear about Image Arithmetic?

Comments:

Future Releases - Mailing list

If you want to be informed of new releases in the future, please send email to the author
Put ***SUBSCRIBE Image Arithmetic*** in the subject field or the body of your email.

If you want to be removed from the list, please send an email and put ***UNSUBSCRIBE
Image Arithmetic*** in the subject field or the body of your email.

Bug Reports

Image Arithmetic has been tested extensively. However, if you find a bug, especially one that can be reproduced, let me know!

Report bugs to [the author](#)

Acknowledgments

All registered trademarks used in this documentation are registered to whoever owns them.

The JPEG coding in Image Arithmetic is based on work of the Independent Jpeg Group.

Thanks to Jon for testing several beta-versions of Image Arithmetic!

The Author

I can be contacted at:

Richard van Paasen
Moreelselaan 42
5643 RN Eindhoven
The Netherlands (Europe)

www: <http://huizen.dds.nl/~buddha/>
email: rvpaasen@dds.nl

