

Scan slides

1.6.21

Table of Contents

1	Preface	1
1.1	nomenclature	2
2	Other sources of information	2
3	Setup	3
3.1	Mechanical setup	3
3.2	Electrical setup	4
4	Photographing with dias.py	5
4.1	Parameters in the dias.py Script	5
4.2	Camera mirror	6
4.3	Camera setting	6
4.4	Camera and projector specifics	7
4.5	Here we go	7
4.6	possible problems	7
4.7	result	7
5	General information about PostDiaDigit	8
6	Rename with Edit / rename	8
6.1	Select / reload directory	8
6.2	Number of slides	8
6.3	Mask	9
6.4	delete	9
6.5	rename	9
6.6	Check and go	9
6.7	save settings	9
7	Cutting	9
7.1	Basics	9
7.2	Cut with the Edit / cut	10
7.3	Problem solving with the crosses	11
7.4	And further	11
8	Duplicates	12
8.1	Use of Edit / Duplicates	12
9	Slide frame	13
10	Feedback	13
11	Legal and source code:	14
11.1	License	14
11.2	source code	14
12	Changes	14
13	Known issues and prospects	14

1 Preface

There are few alternatives to digitize slides, either you have it made, you have to hand in all slides somewhere and pay a lot, or you use a slide scanner, as far as I know there is only one where you don't have to insert each slide individually somewhere. It is very expensive and, above all, like my flatbed scanner with slide holder, it takes forever, depending on the resolution, several minutes per image. With my almost 30,000 slides, that was out of the question. The only method that seems acceptable to me is to photograph the slides in such a way that an entire tray automatically runs through at the push of a button in about the same time the above-mentioned scanners need for a single picture. There are already a few pages on this topic that describe how to put the whole thing together. However, I haven't found anything on the question of what to do with the pictures after they are on the camera's memory chip. It doesn't help if you need a few seconds to take a picture of each slide, but then several minutes for post-processing. Therefore I have devoted most of this document to this topic and have also written a program that is available free of charge including source code.

1.1 nomenclature

The programs were originally created in German, so there may be some corresponding German names. But most of it should be translated. In the instructions I use a few expressions that should be clarified:

- Series: All images that were taken in one piece without changing the position of the camera or projector, which means that they can be cut with the same clipping mask. This can be several trays if you change them with reasonable care.
- Trip: I had mainly travel pictures, but what I mean are simply pictures that belong together, i.e. a trip, a celebration, another event, ...
- jpg: The primary image format used for editing. This can also be another format, but must then be entered in the PostDiaDigit main window and must of course be supported by the QT library used look [here](#).
- nef: The second image format, it is treated parallel to the primary when renaming, if there is a file with the same name with this extension, but otherwise ignored. It must be entered accordingly in the main window of PostDiaDigit, so it can also be set as desired. It doesn't even have to be an image format.

2 Other sources of information

The links mentioned here are only in German, sorry, but there are online translators.

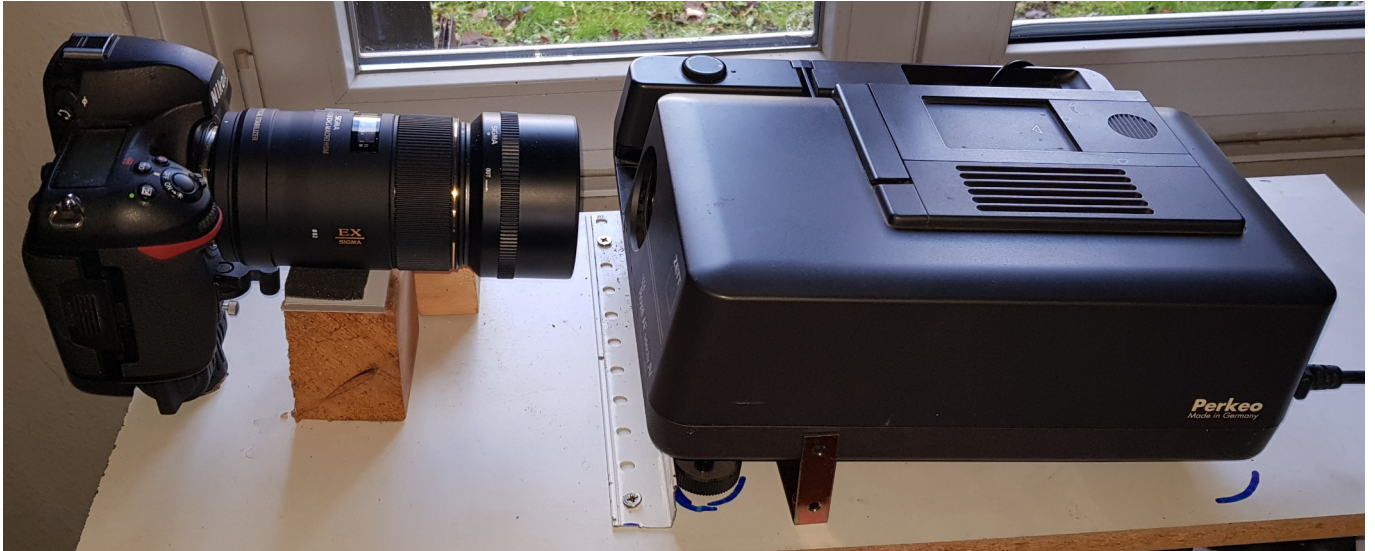
Several years ago I planned to digitize my slides and bought a conversion kit from [fotonovum](#). What I was missing at the time was the right camera.

There are some other sources that show similar structures and from which I was also inspired:

In the [Make](#) Magazine it was described in great detail, the main difference to my solution is the method of switching the projector and the exact setup.

A detailed documentation is available for download on their website for the [AZ-Delivery](#) relay used.

3 Setup



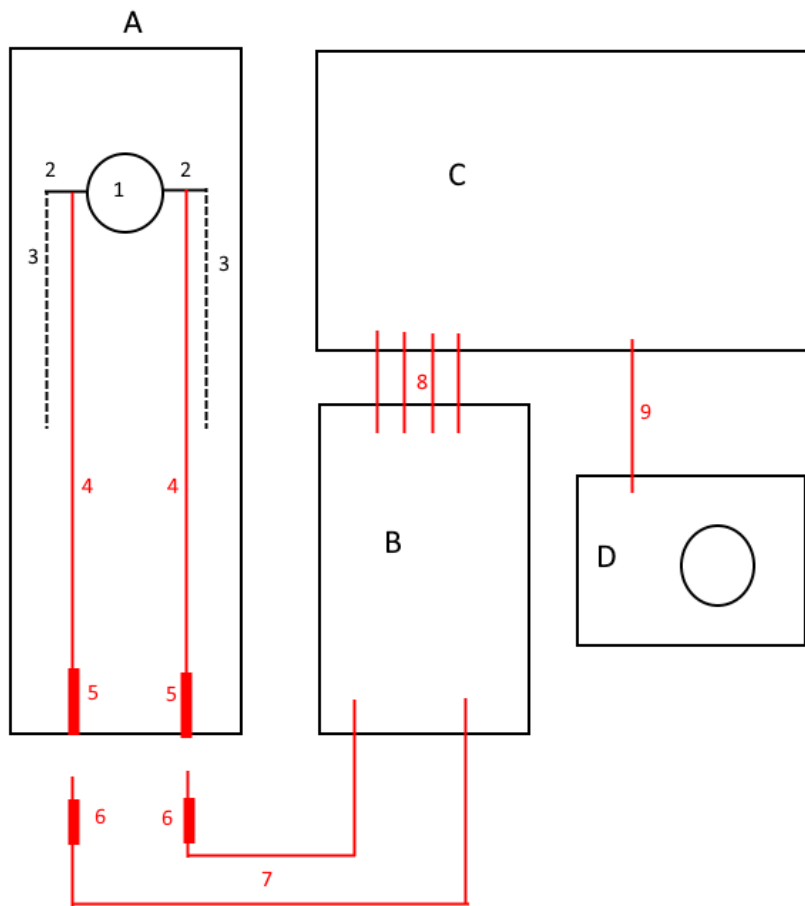
3.1 Mechanical setup

Here you can see my setup, but it doesn't have to be exactly like this, it is only important that the camera and projector are arranged on one axis. The camera is held by the upper part of a shoulder tripod, but since this gave way under the weight of the lens, I put a movable block of wood underneath. Because the tripod head made the camera higher than the projector, I arranged both at an angle. This had the advantage that the lens could be moved up and down a little by moving the block. The two angles (white and gold) serve as a stop and define a fixed position for the projector, but it can still simply be removed. In addition to the common axis, it is important that the slide tray has space, i.e. that it does not hit the lens or the camera. Depending on the geometry, this results in the minimum distance between the projector and the camera. And from this then the optimal focal length. For me, the 105mm macro worked well. Tests with a different lens and close-up lens were not really good. The second block is used to support the slide tray, because the projector sometimes blocked. But that shouldn't be necessary for other projectors. As I said, I used the conversion kit from fotonovum, but with the original lamp. Alternatively, you can cut the right screen yourself, E.g. made of 0.5-1mm thick Teflon, see the above article in the Make. This became necessary for me in the

meantime because the fotonovum screen broke. Overall, if possible, build something from what you already have.

3.2 Electrical setup

I used a Raspberry 4B to control the whole setup. But a smaller one would do it too. And those who are more familiar with electronics can also do something with relays and the like. Essentially you need something that can control both the camera and the projector. If you take a Raspberry like me, you still need relays, I used one from [AZ-Delivery](#).



Legend

- A: remote control for the projector
 - B: Relais
 - C: Raspberry Pi
 - D: Camera
 - red: new cables
 - 1: Button
 - 2: existing connections on the button
 - 3: other existing conductor tracks,...
 - 4: 2 new jumper cables in the remote control
 - 5: female plugs of cables 4
 - 6: male plugs of cables 7, these fit into 5
 - 7: 2 jumper cables to the relay (to "Common Pin" and "Normally Opened pin")
 - 8: 4 jumper cables between relay and Raspberry.
 - 9: USB-Cable
- The designations used in (7) correspond to those in the instructions for the AZ Delivery relay on page 4, for details on cables 8 see ibid on page 14.

To switch the slide projector, I opened the remote control and soldered cables (4) to the contacts of the button, which I then switched via the relay. At first I planned to go directly to the plug of the remote control, so no changes to the remote control would have been necessary, but then I saw the circuit diagram of the remote control by chance. The two contacts are not simply connected, there is something more to it. So the alternative was to recreate it or just solder cables in the remote control. One could do the same with the camera, which was also the reason why I had bought a double relay. But then I found the [gphoto2](#) Lib, this allows you to control many types of cameras simply by using a USB cable (9). This should be more risk-free. How to connect the relay to the Raspberry or similar is well described in the associated manual. It is only important that the pin number used is the same as in the Python script. As in the manual, I used GPIO14 = pin 8. Speaking of soldering, I wouldn't solder on the Raspberry. So-called jumper cables are used to connect this to the relay. I also made the sockets in the remote control of the projector from two female jumper plugs(5): I simply drilled a 3mm hole for each, filed the square plug a little round until it fit into the hole and glued it in place. This is significantly smaller than, for example, a jack socket. Then solder the cable to the button (1) contacts inside. Now you can connect the relay to the remote control from the outside with two more male jumper plugs (6). Otherwise, the two small holes are not noticeable, so you can continue to use the remote control as normal.

4 Photographing with dias.py

This was only my second Python script, so it should be easy for others to make changes, especially since it is a pure text file. Other times, other pin numbers, other gphoto2 commands would come into question ...

4.1 Parameters in the dias.py Script

After the first block with import instructions etc. come the settings:

```
#Hardware
Druecker=8          #GPIO Pin

#Einzelauslösung oder Serienbildfunktion
Serie=1             #0: Einzelbild; sonst Serienbildfunktion

#Zeiten
Wait_t=4            #Gesamtzeit pro Bild
Wait_l=2            #Wartezeit von Auslösen der Kamera bis Transport des Projektors
Press=0.3           #Dauer des Drückens für Transport des Projektors

Denkp=5             #nur bei Serienbild: Anzahl Wiederholungen am Anfang
suchStr=b'Nehme Bild'

triggerCommand = ["--trigger-capture"]
```

This is all German.

Here in detail:

- Druecker is the number of the GPIO pin used.
- TriggerCommand specifies with which command the camera should be triggered with. This is not used with the continuous-advance mode variant. As an alternative to --trigger-capture, --capture-image could also be used. Not every camera supports both.
- Press: specifies how long the contacts of the remote control of the projector should be connected. My projector either goes forwards or backwards depending on the duration.
- Serie: if this is not 0, the continuous-advance mode version is used. Depending on this, the remaining variables have slightly different meanings:

Variable	meaning	Serie=0	Serie !=0
Wait_1	Waiting time from triggering the camera to transporting the projector	Waiting time after executing the TriggerCommand	Waiting time after the camera announces that it will take the next picture
Wait_t	Total duration per image	(Wait_t-Wait_1) is the waiting time after the transport has been triggered. The total waiting time is a little longer, namely: Wait_t + Press + the duration of TriggerCommand	This is transferred to the continuous-advance mode. If the camera manages it, it should be the time between 2 photos.
Denkp	Specifies how many pictures are taken at the beginning of a Tray without changing the slide. See next section	Will be ignored	Is used
suchStr	Search string for continuous-advance mode version	Will be ignored	Part of the feedback from the camera that is searched for for synchronization

4.2 Camera mirror

Since I don't think it is good for the camera if the mirror goes up 30,000 times, I looked for a solution to prevent this. Especially since I borrowed a D850. This is not universally suitable for all cameras. Therefore there are two variants of the script. The universal one triggers the camera for each picture individually. This makes synchronization easy. The alternative version triggers a series of pictures and tries to synchronize the transport of the projector with it. This is theoretically still relatively easy because the camera announces every picture, but by then it is actually too late. So I wait for the message and trigger the image change 2s later, in the hope that the image will be taken by then. The script must of course recognize this feedback. It certainly depends on the language set and most likely on the type of camera. So you have to look in the log file to see what the response is and define part of it as suchStr, see above. Instead of the set 4s, however, the first pictures often take much longer. Sometimes a minute. There were always problems with the synchronization. At some point I gave up and just canceled the first 4 transport orders. The first slide of each tray is therefore recorded 5 times. This can be set via the parameter Denkp in Dias.py, see above.

4.3 Camera setting

I have had bad experiences with all automatic functions. So I switched off the autofocus and also set the exposure to manual. I have worked with the normal projector lamp, others recommend a weaker one. So I used the following setting:

aperture: 22

time: 1/500 s

ISO: 80

With such a small aperture, as was explained to me too late, there is a risk of diffraction blur. Yes, as a physicist, I should have known that ;-)) I did not observe this, but it may be stronger depending on the lens. The advantage is the high depth of field, so you don't have to focus as precisely and it is also forgiving, for example, dented slides. Because of the vibrations in the projector, a short exposure time is certainly good and that the quality is better with a lower ISO number should also be clear.

4.4 Camera and projector specifics

The script has only been tested with a Nikon D850 and a Zett projector. With other devices it may be that you have to adapt something to the synchronization. The corresponding constants are defined at the beginning of the script and can be changed, [see above](#). Depending on what kind of trays you are using, you should also adapt the values offered for the images / tray.

4.5 Here we go

You insert the tray so that the first slide is in the projector. After starting dias.py you will be asked how many pictures should be taken. Some standard values are offered. But there are two special cases:

- 1: only one picture is taken without any transport, without synchronization, ... so only as a remote control for the camera
- ?: You can then enter any value here, e.g. if the tray isn't full.

After clicking on the corresponding button or after entering the number (at ?) Everything goes automatically. The cycle repeats itself until you choose Abort.

The cutting of the pictures (see below) is based on the fact that many pictures were taken under the same conditions, so you should try to take at least 100 pictures in a series without moving the camera or projector. I took mostly 400-500. Then the images must be copied to the PC for further processing. And as always, I recommend making a backup. I also recommend writing a text file in the folder in which you copied the pictures, which contains notes about what you have recorded there, at least if you do not intend to rename it immediately. So it would be important which trays and how many pictures per tray you have taken.

4.6 possible problems

It can of course happen that the projector jams, especially since nobody has a new one. So be ready to intervene if necessary. Aborting the script is not implemented, it is faster to turn off the projector anyway. The synchronization is not 100% accurate. So it happens that one slide is recorded twice and the next one is not recorded. As I said, this shouldn't happen with the single trigger script. I wrote a program to search for duplicates, see below.

4.7 result



Here is an old picture from Berlin. It's upside down and mirrored. It is important for the automatic cutting that there is a black border around the image. The bright spots further out, they are reflections on the tube of the projector, do not interfere. One might think that there is a lot of waste because there is a lot of black all around, but with an picture in portrait format there is only a little

above and below. So if you don't want to turn all the pictures first, you have to live with this waste. If you do, let me know, because then I have to adjust the cutting program, it is currently always looking for the "cross" see below.

5 General information about PostDiaDigit

Originally there were 4 programs that I then combined into one, each of the original programs has an entry in the Edit menu. In the main window you can make some general settings:

variable	meaning	Meaningful values
language	The language of the program.	You have the choice between German, English and French
Primary file format	The primary image format used for editing.	This format must be supported by the QT library used, see here .
Secondary file format	The secondary image format that is only used for renaming.	Any format, it doesn't even have to be an image format

The actual functions can be accessed via the Edit menu. You can also display this help and the copyright message.

6 Rename with Edit / rename

The camera simply assigns sequential numbers with a camera-specific prefix. It makes more sense to give the pictures names that contain the following information:


- Identifier of the series, e.g. a trip
- Number of the tray within the series
- Number of the slide in the tray

At first I did this with [TotalComander](#), but even if it is well suited for this, it was still a lot of work, so I wrote dias_rename, which later became part of the PostDiaDigit program. In addition to the actual renaming, the multiple slides at the beginning of the tray must also be sorted out. The program only renames, it does not delete, but also does not copy anything. So afterwards there are the same files as before, but sorted into subfolders and with new names. It assumes that you took RAW and JPG images. These are moved to 3 automatically generated sub-folders:

- del: this is where the duplicates end up and can therefore be deleted by the user
- jpg: this is where the jpgs end up
- nef: guess what

The extensions jpg and nef can be set in the main window. The primary ending is that you are looking for, e.g. the jpg. After moving the jpg, a check is made to see whether there is a file with the same name with the other extension, if so, this is moved / renamed accordingly. This is important if you intervene by hand in between. If you have garbage, for example, you can simply delete the jpgs, or move them to the del folder, the nef of the same name then remain at the end. After such an action, the folder must be reloaded, see below.

6.1 Select / reload directory

With the button “...” you can load a new directory, with the other “” you reload the current directory, useful if you have deleted something by hand or changed something in some other way.

6.2 Number of slides

After 2do it shows how many pictures are still available, so only the jpg!

Then you can choose how many pictures are in the tray. You can simply select some standard sizes, for everything else there is the “?” And an input option behind it. A special case is the “1” here the first picture is renamed and nothing else, so nothing is deleted or renamed. Of course, at most as many pictures are renamed as there are still left. So you don't have to reduce the number for the last tray in a series. But if a tray is incomplete in between.

6.3 Mask

The filename is composed of the following parts:

- Prefix: any text
- Number of the tray, always at least two digits. If you change this, the slide number is set to 1. This has proven to be useful in practice.
- A separator, although editable, should remain “_”, especially if you also use the frame program.
- The number of the slide in the tray. For trays with more than 99 slides three-digit otherwise two-digit. Here you can enter the start number if it should not be 1.

6.4 delete

As I said, it is only moved to the del folder, not really deleted. This is necessary because the camera I used had synchronization problems at the beginning of a tray. The preset is accordingly that images 2 to 5 are deleted. The reason I don't delete 1 through 4 is very simple. The slide gets warm and can then possibly deform ("jump") so I use the coldest one. If you have recorded a series without duplicates, enter 0 as the first value, then nothing will be deleted. Then it will be shown which images will be deleted, i.e. the original file names.

6.5 rename

The original names of the first and last images that are to be renamed are displayed here.

6.6 Check and go

I recommend to have the directory open in an Explorer window, then you can easily check whether everything is correct. In order not to see jpg and nef one after the other, it is best to select "Group by type", then it will be clearer. One advantage of the duplicates is, for example, that you can easily recognize the beginning of a tray. So you check if the first 5 pictures are the same and whether after the last picture (according to the rename box) there are again 5 duplicates. If everything is OK, click on “Go!”. Since it is only renamed, this goes quite quickly. The program tries to guess what to do next. If you have renamed a tray, the next one will probably come, so it increases the tray number and sets the start number to 1. If you have only renamed one picture, the start number is increased by one. Ideally, you can just keep clicking Go until the directory has been processed. If there were pictures of different trips in the serie, you might get the idea of separating them now, but you should only do that after cutting.

6.7 save settings

Here the settings which are specific for the respective user are saved in the registry. These are: the two file types (e.g. jpg and nef), which images are to be deleted (e.g. 2-5) and the number of images per tray.

These settings are loaded automatically when the program starts.

7 Cuting

7.1 Basics

You now have many images that contain a lot of black and possibly reflections on the tube of the projector and the image in the middle, either portrait or landscape, and they are upside down and reversed. The photos have to be cut to size and mirrored, and it has to be recognized whether they are portrait or landscape. All of this as automatically as possible, as safely as possible, so that nothing is cut away and everything as quickly as possible. There are two ways to do this:

- Analyze and cut each image individually. This will fail for images with dark edges and take a long time.
- Use the same two clipping masks for all images in a series.

I decided to take the last option. Essentially I first superimpose a lot of pictures, usually 100, then you get

a light cross on a black background. The program can then determine the clipping masks from this. These will not fit 100% for all pictures, as not all pictures were photographed exactly the same. The reasons are:

- different frames
- The mechanism of the projector will not work the same every time, to a fraction of a millimeter.
- The camera or projector was moved. So avoid this as much as possible!

But chances are bigger, that there will be a black border left than that something is cut away. If you want to view the pictures later, e.g. on the TV or PC, you usually have a black background, then this error is not noticeable, the picture is only displayed slightly smaller than it could be. Experience has shown that 100 images are enough to calculate the best possible mask, with my 28,000 slides I had to increase the number for just 1 serie. More pictures make a better mask, but of course take more time.

7.2 Cut with the Edit / cut

With the button "Settings" you can change some settings. Apart from the language, you should ignore this first and try to see if it works with the default settings. If not here the possible settings:

Variable	meaning	Meaningful values
Number of images for clipping mask	The calculation of the clipping mask is quite time-consuming, but if you use too little, the mask becomes bad.	I used 100, I only had to increase it once.
Ignore small angles	When determining the clipping mask, it is also recognized whether the camera and projector were rotated against each other. The program can correct this, but of course it is better to correct the error mechanically. Correcting small errors is not worthwhile, so you can enter a threshold here.	I used 1 degree.
Outline length	When searching for the "cross", other spots will of course also be found. However, these are usually much smaller than the cross. The most important criterion is therefore the length. If you enter a negative value, the image width is used.	If the cross fills almost the whole picture, the default setting -1 is ok. Otherwise look how wide such a cross is and multiply the value by 4 and round off a bit.

The above settings are saved in the registry with the "Save" button. These settings are loaded automatically when the program starts.

The program cuts all images of the primary type in a directory that can be selected with the "Select directory" button. All images with this extension in the folder will then be processed. In the best case, you will find all the images cut later in the folder of the same name. However, some of them almost always end up in the "portrait" folder. The logic is as follows:

- the picture is identified as portrait → it is cut as portrait and written in the "cut" folder
- the image is identified as landscape → it is cut as landscape and written in the "cut" folder
- The program is not sure and creates both versions, the landscape is saved in the "cut" folder and the other in the "portrait" folder.

So now natural intelligence is required. You look at the pictures in "portrait" and do the following:

- it is an portrait picture → leave it there
- it is a landscape picture, so because of the wrong cut you can only see the square part in the middle → you delete it.
- it's black → you delete it and write down the number to clarify what happened there, e.g. whether there was a gap in the tray or there were problems with the recording and you have to photograph it again.

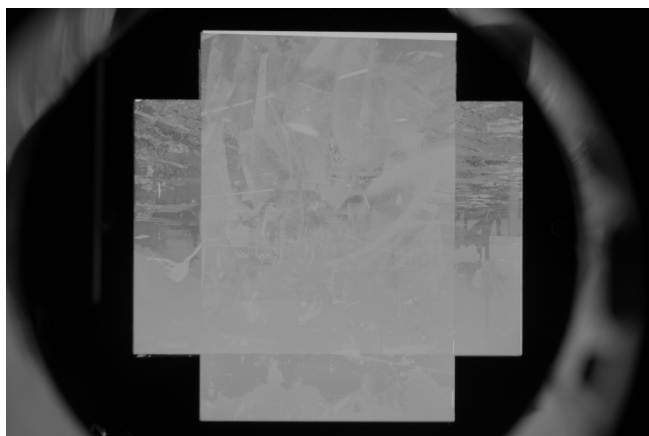
Then you move all the leftover images from "portrait" to "cut", which means that all portrait images that were not automatically recognized as such replace the wrong landscape versions. Now you have all the correctly cut ones together, plus the black ones. If there are gaps you delete them, otherwise just take an-

other picture. If up to a certain point there are few pictures in "portrait", but from then on almost all of them, it can be that you somehow moved the projector or the camera. Then the best way is to divide the images into two series and to cut both separately.

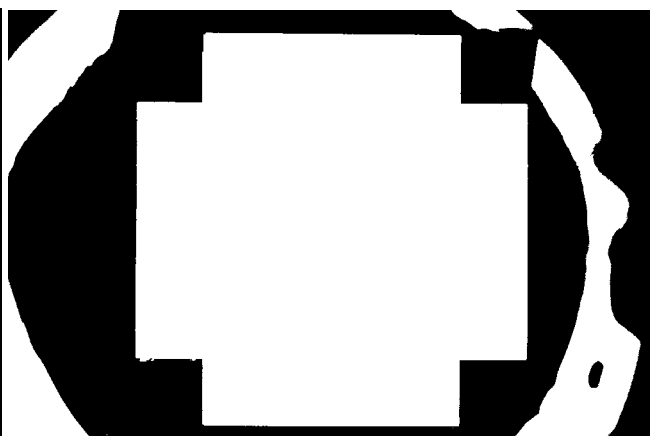
7.3 Problem solving with the crosses

In rare cases, the program reports that it thinks the clipping mask is bad. Then you should first take a look at the images created with the name Kreuz (Cross in German). There are 3 of them:

- Kreuz_Grau.bmp: the overlaid images, even with 100 overlaid images, you can still see something [here](#)
- Kreuz_m2.bmp: m like monochrome and 2 because the resolution is doubled. The example below is the ideal case, a very sharp white cross on a black background.
- Kreuz_u2.bmp: Only the outlines with double resolution. To show such a picture here as a whole would be pointless, it is almost completely black, so only a small section (about 1/40 of the width) is shown.



Kreuz_Grau



Kreuz_M2



Excerpt from Kreuz_u2.bmp (top right corner)

Depending on what you see there you have two options, either you increase the number of images from which the mask is formed and try again, or you adjust the cut.csv file by hand and start the process at step 2, the actual cutting, by using the "cut.csv" button. If you have the coordinates from the crosses with a 2 at the end, you have to devise them by 2!

If the angle of rotation is anything > 1 degree (see settings), you should check the alignment of the camera / projector. The program then rotates the images, but this takes time and does not improve the quality.

7.4 And further

Now is the time to sort the images the way you want them. Since all the pictures were taken with the same exposure, they still have to be corrected. For this I used [Irfanview](#)'s batch mode (auto-correction). This is also the reason that the cutting program does not generate jpg but png files, jpg is lossy, and you shouldn't have that several times in a processing chain. This would of course speak in favor of not using

the jpgs of the camera but the raw images at the beginning. But that would be complicated and I was happy with the result. Complicated, since the QT libraries used cannot read any nef. One possibility would be to use a RAW converter to convert the images into a supported lossless format such as png. The main reason to save and rename the raw images is that you may want to process a few images in better quality, not just for the slide show.

8 Duplicates

As described, synchronization problems can lead to duplicates, i.e. two images of the same slide. Finding this is particularly important, as there is usually a different slide missing for each duplicate. I noticed this problem quite late. Since I didn't feel like looking through the 28,000 slides, I quickly wrote a program that makes a preselection. Fast means it is written quickly but not very optimized. For a more optimal program I would certainly have needed hours of programming time, but only saved significantly less than an hour of time in use. The aim was to show the user the smallest possible number of suspicious images that contain as many duplicates as possible. So be more tolerant and quick when making comparisons. One picture is always compared with the next. The result is a two-stage process:

1. almost identical images should be almost the same size as jpg. If it is less than 2% different, they are suspicious. This comparison is very quick because the images do not have to be loaded
2. Now they need to be loaded. Since 2 photos of the same slide are never exactly the same, it is not possible to simply compare them pixel by pixel, at least not in their original resolution. I reduce them to $3 * 3$ pixels and then compare their color values.

Instead of going through both steps, which would take some time, the first suspicious pair of images is displayed after step 1. While the user is looking and comparing, step 2 is applied to the next images in the background. The number of remaining suspects is shown below the images. But don't get a shock, these are all those that are suspicious after step 1. If you click on the ">>>" button, this number is usually reduced by more than 1 and you can see the next 2 pictures. The path is also displayed. After step 1, a few percent are suspect, after step 2 a few per thousand. So even with my number of slides, I was through quite quickly. I found 10 duplicates. So it's not a really big problem.

8.1 Use of Edit / Duplicates

With the "..." button you can select a directory. Then the program starts. It searches all sub-folders for images, except for folders with the name Docu (which are reserved for the frames). After a while, the first two images appear. If they are the same, write down their path. Then click on ">>>" to see the next pair of images. This can take a while.

9 Slide frame

So now we have the slides as digital images, but something is still missing. Many will have labeled their slides like me. What do you do with it now? Basically there are at least three options: typing, scanning or photographing. Unfortunately, all three have a serious disadvantage, you have to pick up each slide individually, something that I actually wanted to prevent. My method was to put the slides on a slide lighting desk and take photos. With normal trays this happens very quickly, with LKM or CS trays it is unfortunately a lot of fiddling. But at least something that you can do while watching TV, for example. My desk has space for 40 slides. So you first get a picture like this:



In addition to the slides, the sticker of the box can be seen at the top right and a note with “1-40”. So these are the first 40 slides of the U1 tray. And then there are the three red and the green dot. Instead of this one picture, I wanted 41, so 40 pictures each with a frame and one with the sticker. You can of course create a static arrangement of the desk and camera, similar to the projector and camera, so

position the camera somehow with a tripod above the desk, but I thought that it was too complicated. So I decided that I would just take a picture of it by hand, with my mobile phone and then try to determine the position of the desk. If you have the 4 corners of the desk, it is easy to cut the 41 out of one picture. The 4 colored dots are used to find the corners. If I had known how complicated it is to clearly recognize them, even if the lighting situation is different, there is a slide with a lot of red or something similar, I would have probably chosen the tripod solution, but I had already photographed almost all the frames. I don't think the program is in a state to be suitable for the general public. It only works with my geometry, with points of this size and color ... As a compromise, I included the routines in the program and bundled the many constants that depend on the specific geometry. If someone wants to use it, he should contact me. Then I will think about how to adapt the program to the probably different geometry and will expand this guide.

QR codes might make more sense than the colored dots if you can find a library that can find the position of these in an image.

I also have a program that creates an html in which you can display the appropriate frame by entering the time, in addition to information that you have entered as text. Provided, of course, that you have a slide show in which each image is displayed for the same length. My slideshows start with an introductory of 16s and every picture is displayed for 7s. So if I'm at 1:30 I'm at picture $(90s-16s) / 7 = 11$ (rounded up). So the 11th frame is displayed ... But even there a lot of work would be necessary to adapt it for normal users. I would only do this if there is a corresponding interest.

10 Feedback

I only assembled the PostDiaDigit program from the individual programs after I was finished, so I never used it except for test purposes. In the meantime a friend is using the setup, but with the same projector and the same camera (it's his ;-)) The program was created to work with my constellation. Of course I always tried to keep it as variable as possible. Whether this is enough will only be seen when others use it with different hardware. So I would be grateful for further reports using different hardware and the necessary adjustments. If the effort is reasonable, I am also happy to make adjustments. Possible

problems would be, for example, if there are significantly more (or less) portrait or landscape images, or even only one of the two.

11 Legal and source code:

11.1 License

This program is under the [GPL V3 license](#).

A copy is enclosed with the program.

11.2 source code

The source code should be available where the program is available, but at least on my [homepage](#).

12 Changes

13 Known issues and prospects

As I said, the program and the documentation will only be further developed if there is corresponding feedback, because I have finished editing my pictures.

Dimitri Junker