

# Finding the nodal point

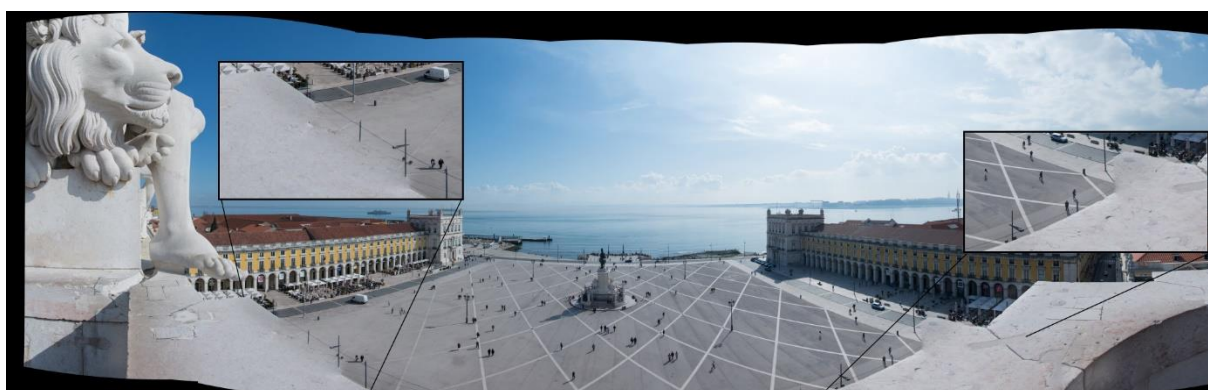
A simple hands-on instruction using PTGUI

By Frank Cullmann  
Culafra@googlemail.com

*This article wants to guide you to a new method of finding the nodal point (no-parallax-point) of your camera-lens combination. It may be a little bit nerdy, but it is straight forward.*

Making a panorama with objects being quite far no special care has to be taken regarding the center of rotation of your camera. The fusion of the pictures to a panorama is usually no problem

However, for more advanced panoramas, where there is both a foreground as well as background or even a 360° panorama, you cannot avoid the topic of determining the nodal point. Only then, it is possible to obtain single pictures, where the rotation around the axis does not lead to a shift between foreground and background for the different pictures (parallaxes). This will lead e.g. to “ghost ridges” (picture 1) or breaks in straight lines (picture 2). Although a lot can be corrected via masks in PTGUI and your picture editing software, however this is time consuming and often frustrating.



Picture 1: Praça do Comércio in Lisbon

Luckily, there are a lot of companies selling nodal point adapter, which will conveniently rotate the camera around the nodal point, that is, if you know it and set the adapter accordingly.

Unfortunately, there is no possibility to calculate the nodal point theoretically (at least I do not know of any), but one has to determine it for each combination of camera and lens, for zoom lenses even for the given focal lengths.



Picture 2: Inside the Sagrada Familia in Barcelona

A first, very welcome help is a site with a whole series of single data for cameras and lenses ([http://wiki.panotools.org/Entrance\\_Pupil\\_Database](http://wiki.panotools.org/Entrance_Pupil_Database)). Since those data can be added up, you can calculate the nodal point, if the data of your camera and your lens is listed there. Unfortunately, there are a lot of data for cameras on this site, but only a limited number of lenses.

The most common listed method however is the following experiment (also described on the site <http://www.pixelrama.de/panorama/aufnahme/aufnahme-5.html>). You look for a motive, which has both foreground and background in it and for which you expect some decent parallax error. Then you do a lot of test shots, where you slightly change the position of the camera on the slide of the nodal point adapter. You take 2 pictures at each position with approx. 30% overlap and note the position where you cannot see a difference between the two pictures.

This method seems simple and quite straightforward, however for me it has some drawbacks: First of all, for a standard lens this might work, however for wide-angle lenses and super wide-angle lenses (those you want to use for larger panoramas) I find it quite difficult to see the differing details. Maybe I am optically challenged, but when I decided on the nodal point based on this method (with some guessing), it did generally happen that in the resulting panorama there were parallax errors visible. This was especially true for 360° panoramas.

To cut a long story short: I do not want to determine the point where I cannot see any differences, but the point where there are none.

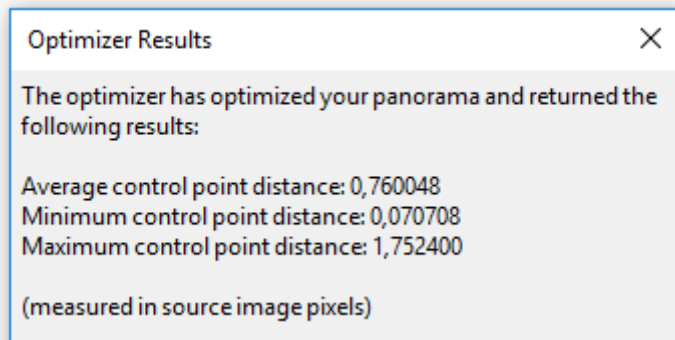
Then I got the following idea: If I am annoyed that PTGUI finds parallax errors, although I had not seen them, then PTGUI should examine the pictures for me.

The method works quite simple as follows:

You take the picture pair as described above, while constantly varying the position of the camera on the slide of the nodal point adapter and note the position for the pair of pictures.

Then you upload the pictures to PTGUI, align the 2 pictures, choose “advanced mode” (in the upper right corner), choose the tab “optimizer” and choose “Run Optimizer” (in the lower left corner).

You will obtain for this pair a number for the different control point distances like shown below.

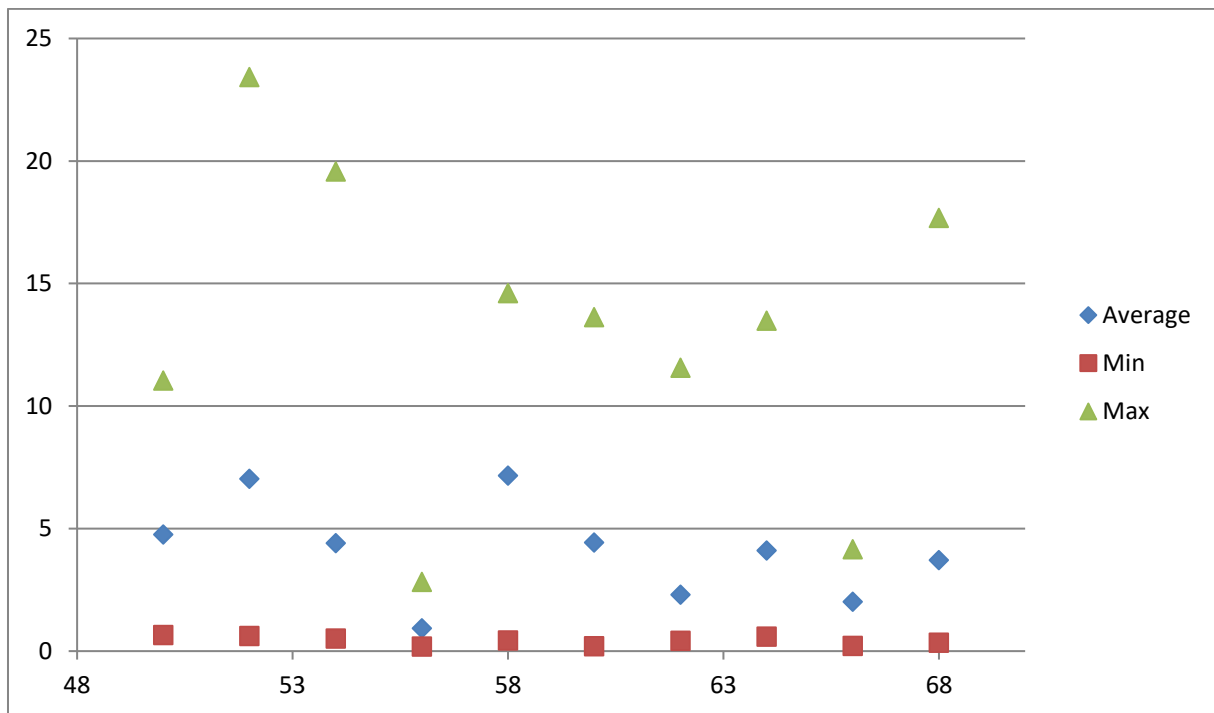


You then note the 3 values and take the next 2 pictures and repeat until you have the value triple for all the positions of the slide.

Then you put these data into Excel e.g.; for my Nikon D7100 and the Nikkor 35 mm f1.8, the table looked like this:

mm	Average	Min	Max
50	4,759	0,661	11,048
52	7,036	0,627	23,429
54	4,41	0,522	19,579
56	0,932	0,193	2,828
58	7,162	0,434	14,608
60	4,435	0,204	13,633
62	2,314	0,427	11,57
64	4,114	0,597	13,489
66	2,026	0,221	4,165
68	3,717	0,346	17,681

Then you ask Excel to convert the table into a graph:



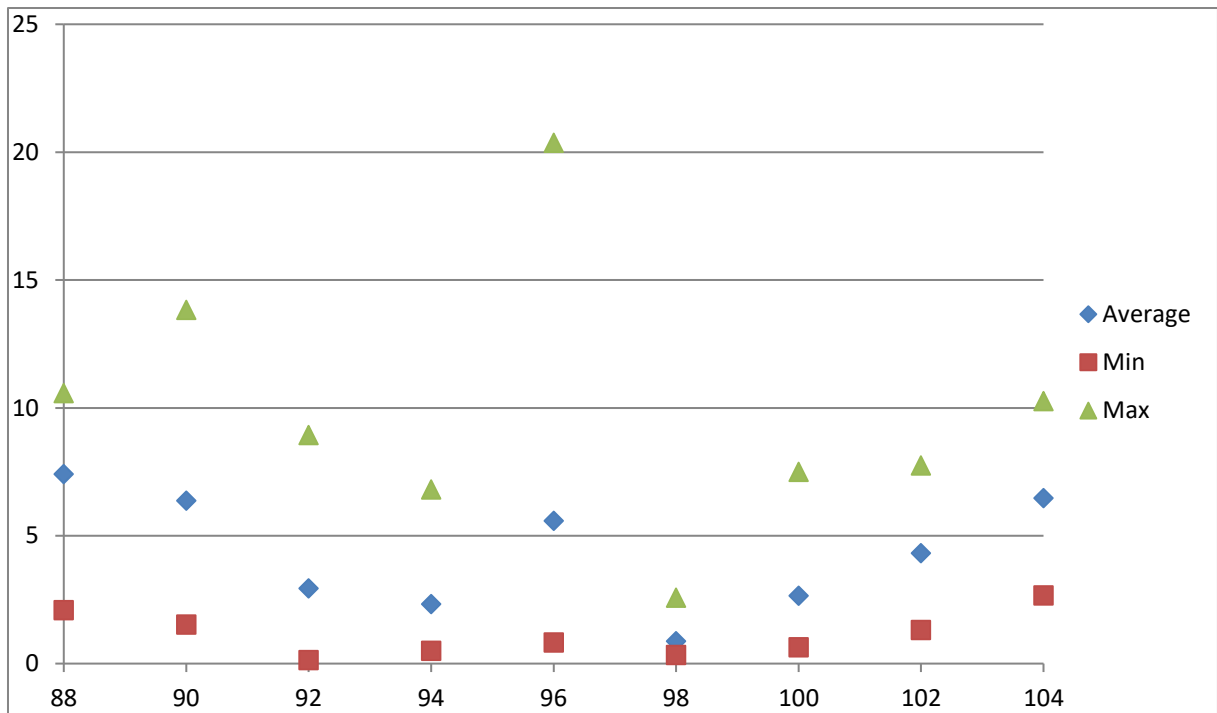
As one can see, this is not an exact science, however based on the Max value, the correct position for the slide should be 62 mm.

Normally, I take a pair every 2 mm. Of course, I could have repeated the process with 1 mm difference around 62 mm, however based on my experience more precision is not needed especially since out in the field I cannot really adjust the slide with higher precision.

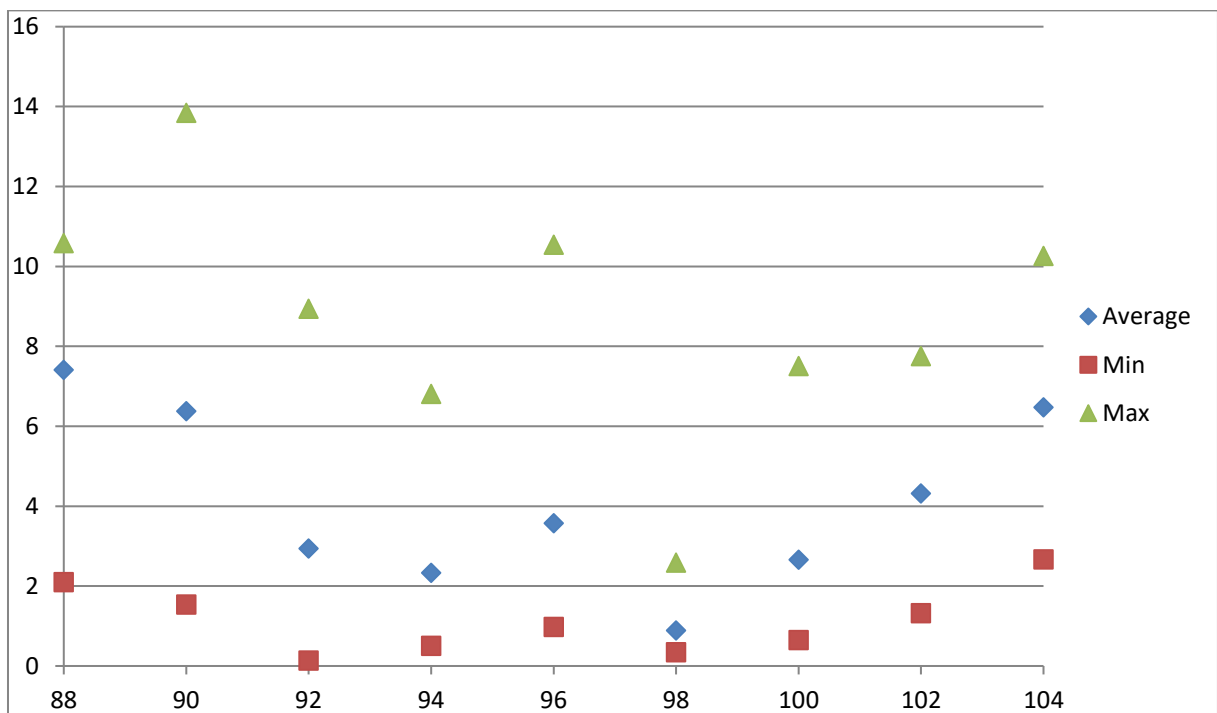
The advantage of this method is the fact that you can systematically analyse the range on the slide you want to cover without readjusting or repetitive going back and forth on the slide.

Sometimes however, some shots (or the whole set of shots) create issues. This should be demonstrated in the following examples.

The first example is with the Sigma 18-50, f2.8 @18mm. I received the following graph:

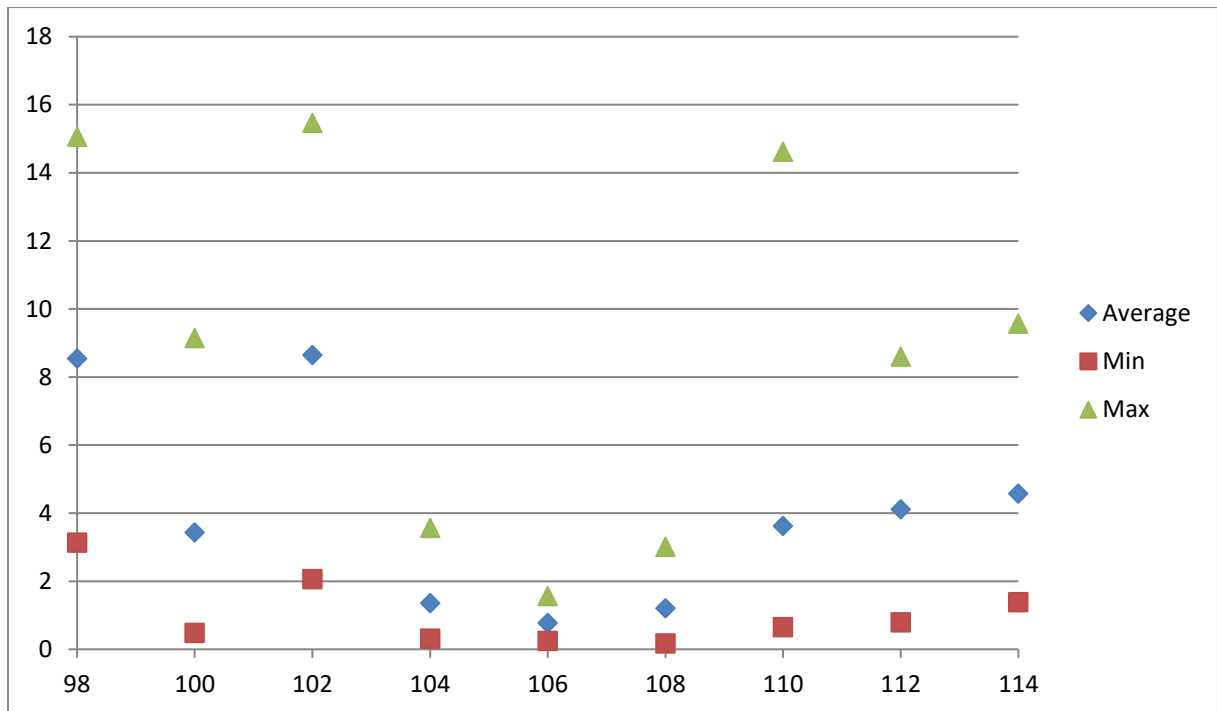


Most probably, the nodal point is at 98 mm, however the values for 96 do not fit at all. I regarded the control point for the maximal distance as an outlier and deleted it and reoptimized the pair of pictures in PTGUI to receive a new data triple. The resulting graph was:

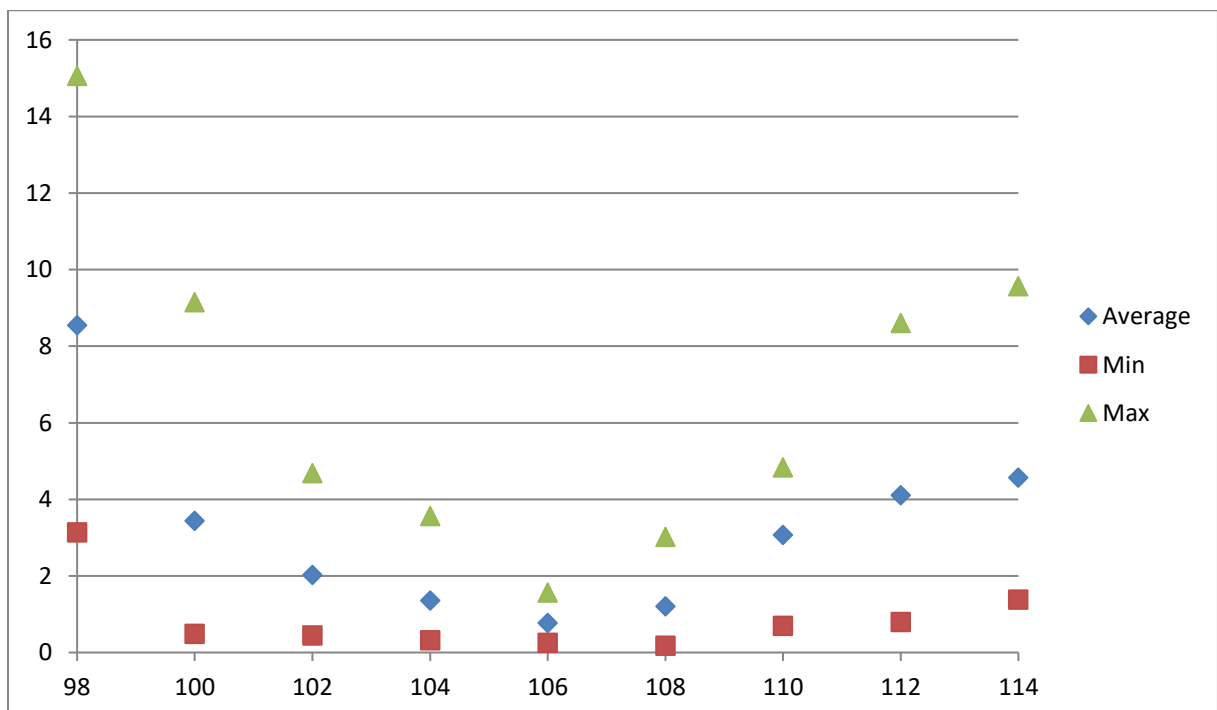


Well, better in general, however still not that satisfactory. I could have chosen to redo the picture shots, but I decided to use 98 mm, since the overall pictures also for the average distance pointed to that number.

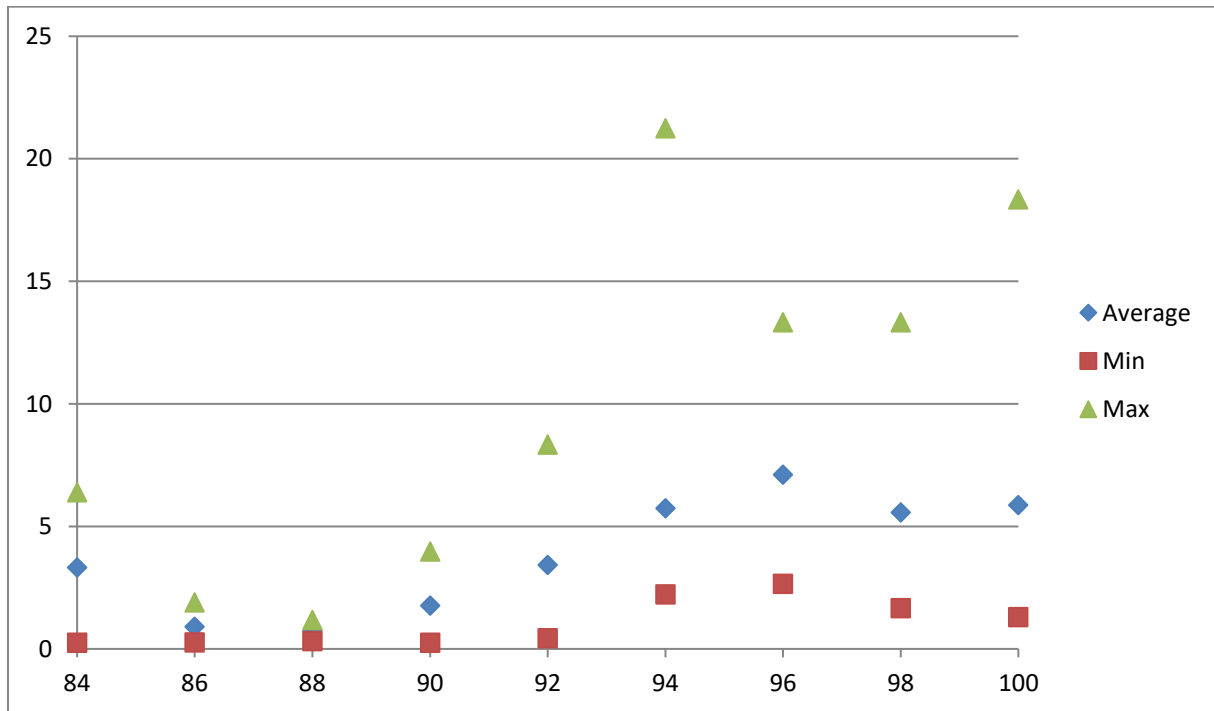
In the second example (Nikon D7100 and Sigma 10-20 f4-5.6 @10mm), this method for outliers worked perfectly. The first result was as follows:



I decided to delete the outliers for 102 mm and 110 mm, redid the optimization and the result was quite amazing:



Now for the last example, the Fisheye Nikkor 10.5 mm f2.0. Fisheye lenses unfortunately do not have a distinct nodal point, but the nodal point is depending from the angle in which the light enters the lens ([http://www.pierretoscani.com/echo\\_fisheyes\\_english.html](http://www.pierretoscani.com/echo_fisheyes_english.html)). For that reason, the nodal point can only be determined for a defined swing angle, in my case, every 60°. Therefore, the following experiment was performed with an angle of 60° between within the pair of pictures.



Based on this graph, I determined my nodal point to be 88 mm. You can see that the area I choose was slightly off to larger values. I calculated from the data of the internet that the nodal point should be at higher values but these data were probably not reliable. I choose 88 mm and the little planets below (pictures 3-5) show that this was no bad choice.

Perhaps you ask yourself why I did also assess the data for minimal distance although they had never any influence on the decision? Firstly, I this knowledge came only after I assessed them and secondly, it's easier to note all three values during this repeated exercise instead of taking care which of the data to ignore.

Overall, this method shows an elegant alternative to numerous other methods described in the internet to determine the nodal point systematically.

  
**FRANK CULLMANN**  
 ————— FOTOGRAFIE —————

<http://www.fotocommunity.de/fotograf/frank-cullmann/fotos/904174>



Picture 3: Kaiserplatz in Frankfurt am Main, Germany





Picture 4: Taunusanlage in Frankfurt am Main, Germany



Picture 5: Town square in Kiedrich in the Rheingau region, Germany