A large format Photographer's *vade mecum*[*]

TI89 program documentation

Robert E. Wheeler

Jan 1998

Contents

1 Introduction

When a university education was mostly a study of Greek and Latin, schoolboys carried about small books containing crib notes, grammatical memoranda, and other helpful bits of information. It is not surprising that they referred to them by a Latin tag, *vade mecum*, which translates as "go with me." I, and I think most large format photographers, do something similar. In my bag is a notebook filled with various things, such as filter factors, reciprocity times, shift limits for various lenses, etc.

In addition, I carry one other thing, which I consider my real *vade mecum*. It is a pocket calculator, and in it I have programmed those calculations that are useful for my sort of photography. Yesterday, for example, I took two pictures while on a walk. One required a tilted lens, and the other involved a nice calculation of DOF to include an object in the foreground. Neither took more than five minutes, and a third photograph was abandoned before the camera was unpacked, because preliminary calculations indicated that it could not be taken in the way I envisioned — I'll probably go back and rethink the composition sometime.

When I started in large format photography, I was unable to find adequate instructions for many fiddling problems in the books I consulted. I found exhaustive discussion of light and photographic materials, but little help on the practical problems. Lens tilt, for example, seemed as much a mystery to many authors of photography books as it did to me — at least if they knew its incantation, they chose not to reveal it.

As near as I could make out, tilting the lens was something to be done by cut-and-try — focus on something, tilt the lens a bit, focus on something, adjust the tilt, etc. until all parts of the subject are in focus[1]. I tried this a few times with middling success, but found it hardly a satisfactory procedure: if for no other reason, than that it takes a long time. I am sure the information exists in the technical literature somewhere, but in exasperation I sat down and derived the equations from first principles. By that I mean I went back to a theorem of projective geometry due to Desargues which underlies the rules artists use to produce perspective drawings, and derived the lens equation relating the focus distance, the lens-film distance, and the lens focal length. From this everything else follows.

[1] A discussion of this procedure in exquisite detail may be found in Bond (1998).

The programs described here, calculate the lens tilt angle in several ways. The easiest is by focusing on two points in the subject plane, which is the idea that underlies the Sinar, and Linhoff built-in calculations. Almost as easy, is to use the distance and angle of a pair of points in the subject plane. There are two additional methods which may appeal to some who fancy that they can judge angles by eye.

Of course, a photographer has other problems: depth of field, DOF, for one. The standard formulas that appear in books work well enough for intermediate and distant objects, but are considerably in error for macro photography. This is strange, since the correct formulas are not difficult. Perhaps the authors have stuck to formulas that they thought would be easy to calculate. In any case, if one is going to use a calculator, which I hope you will, then there is no need to use the wrong formula. In addition, the practicing photographer needs to make judgments about DOF for tilted lenses, and assess the degree of blur for a background or foreground object. Six programs are provided for these problems, differing in the type of input required.

Various questions arise in practice about focal length: (1) given near and far objects, which focal length will cause these to be the near and far DOF limits? (2) Given the magnification and the distance to the subject, what is the proper focal length to choose. Similar questions occur in relation to the proper f-stop. And what about the bellows factor? There are other questions still – what shutter speed will stop motion, or what is the angle of view? Programs are provided for all.

Perhaps the most important thing, is the viewframe: a simple frame made from what you will with a measuring cord attached. I am not aware of any serious discussion of this most useful device. It not only concentrates the photographers attention by framing the view, but enables the appropriate lens to be chosen, and with the two programs included, allows the photographer to find the DOF – all without unpacking the camera.

The programs may be downloaded from a PC via a serial cable. Instructions for downloading are given in an associated document. The mathematical details of many of these calculations do not seem to be available in the literature, and so a collection of notes may also be downloaded.

I assume that the reader is a large format photographer, and that the problems discussed will be those that have been thought about. I wouldn't discourage someone who is new to this format from reading this material and using the programs, but I think it unlikely that they will fully appreciate the programs until they have experienced some of them firsthand. There are a number of good[2] books on large format photography, and I would hope that the neophyte would consult them first. Two that I can recommend are: Stroebel et.al. (1986), Stroebel (1993) For the more technically inclined, Ray (1994) and Jacobson et.al. (1988) are also good.

1.1 Notation

I have chosen to use a few standard symbols both in the programs and in this text, rather than repeat descriptive phrases at every point. They are illustrated in Figure 1, and listed in Table 1. The figure shows three principal planes: the subject plane, the lens plane, and the focal plane. In addition, the Near and Far DOF planes are shown about the subject plane, as are their cognates about the focal plane. Note that the Near and Far planes are measured from the lens plane. The distances of these planes from the subject plane are denoted FrontDOF and BackDOF. The Near and Far planes are not symmetrical about the subject plane, but their cognates about the focal plane are symmetrical, for all practical purposes. Thus one may use $\delta/2$ to represent the distance from either to the focal plane. The lens tilt angle, θ, will be discussed later.

In addition, units of measurement are indicated by appending a unit designator starting with an underline "_" to values. Thus 5_m means 5 meters, 3_mm means 3 millimeters, 10_° means 10 degrees, etc.

1.2 Fractional f-stops

The N values are spaced uniformly on the shutter, and one can set intermediate values, such as a value half way between 16 and 22. The corresponding N as a decimal is 19.6, which requires some calculation to discover, and would be a nuisance were the programs to demand its input. Therefore, the programs allow the input of intermediate values using the form 16_.5, where the "_" is required as is a period before the fractional part: 16_.5 may be read as the shutter setting half way between 16 and 22. In actuality, the program

[2] Although on topic, I cannot recommend Merklinger (1992) or (1993) because simple ideas are made overly complicated.

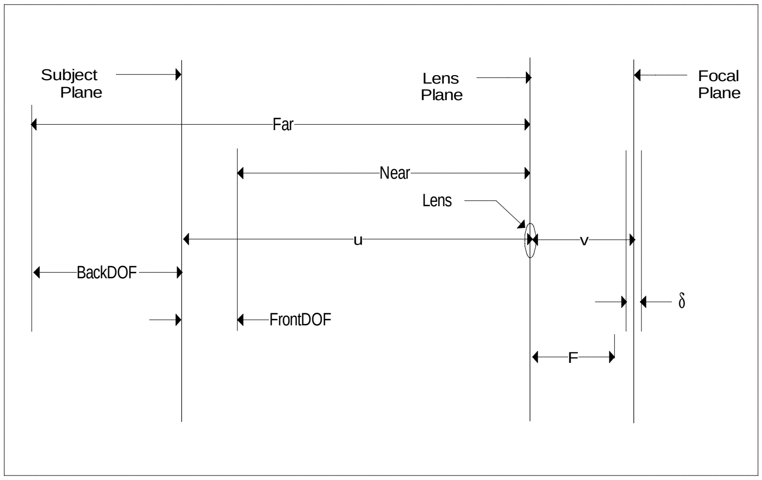


Figure 1: Parallel planes

translates this into a decimal number[3] (19.6) which it stores in the variable N; and when it displays N, it displays a string using "_" if needed.

1.3 Numerical accuracy

All calculations are made with at least 12 significant digits and are rounded to reasonable precision for output display. This rounding may cause the results of two complimentary programs to differ when the rounded output of one is input to the other. For example in Section(3.2) one inputs $u = 10$ m to the program Viewu(), and the program outputs $v = 215$ mm. When one inputs $v = 215$ mm to the complementary program Viewv(), as in Section(3.1), then $u = 9$ m is returned. The corresponding FrontDOF and BackDOF values between the two differ as a consequence. Allowing more decimal values would make the two results agree, but it would complicate things and have little practical effect. The problem only occurs in the output since one may input as many decimals as one likes: suppose one knew $v = 214.5$ mm then one could input it, and the output calculations making use of this greater accuracy, would give $u = 10$ m, and the DOF values would agree more closely.

[3] Actually input numbers are rounded to the nearest f-stop before assignment to the variable N, so inputting either 20 or 20_.5 results in N being set to 12 or 22_.5, respectively.

u	Distance from subject focus point to lens
v	Distance from lens to film
F	Focal length
N	The f-stop
δ	Defocus $\equiv$ Depth of focus
θ	Lens tilt angle
Near	Near DOF limit
Far	Far DOF limit
FrontDOF	$= (u - Near)$: i.e. DOF in front of focus
BackDOF	$= (Far - u)$: i.e. DOF in back of focus

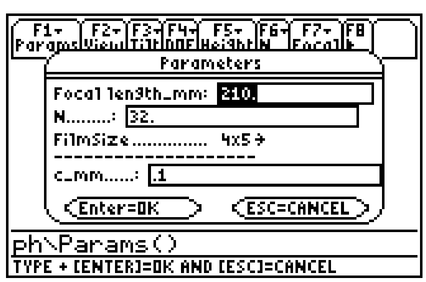
Table 1: Frequently used symbols

2 Setting parameters

Program Name: Params()

Three parameters are needed for most calculations. They are focal length, the f-stop, N, and the format. It is incumbent on the user to make sure that they are always current. Some programs use them, others do not, but it is a waste of time to try to remember which is which, so always make sure they are current.

They may be set by running the Params() program, which is available from the menu. This function throws up a display showing the current values, and allowing them to be edited. The following screen capture shows the display.



The film size is changed via a drop down list. When it is changed, the diameter of the circle of confusion, c, also changes. The changed value will appear the next time the Params() funciton is called. Usually, *c* should not be changed by the user, since it is set at values appropriate for the various formats.

To change the film size, move the cursor to the **FilmSize** row, press the right arrow, and choose one of the options: they are 35mm, 645, 6x7, 6x9, 4x5, and 8x10. When the parameters are correct, press the **ENTER** key.

The c values built in for the various formats are:

Format	c
35 mm	0.03 mm
645	0.05 mm
6x7	0.06 mm
6x9	0.07 mm
4x5	0.10 mm
8x10	0.20 mm

Table 2: c settings

Illustrations in the rest of this document assume the parameters are set as in the above screen capture – Focal length at 210 mm, N at 32, and FilmSize at 45.

3 The viewframe

Even before the camera is set up, a decision about the view to be captured should be made. This is aided considerably by the use of a simple viewframe held so as to frame the scene. Almost anything with the right sized hole will do — I use a bent coat hanger. The inside dimensions should be the same as the size of the image on the film. In addition some means should be provided for measuring the distance from the viewframe to the eye. I attach a flexible tape marked in millimeters to my viewframe. Figure (2) shows some items from my gadget bag, among them may be seen my bent coat hanger viewframe.

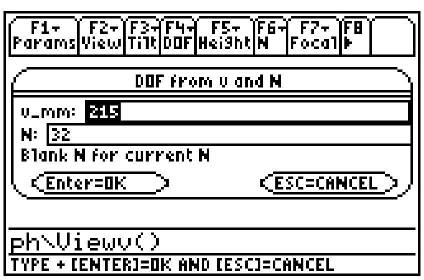
Use the viewframe to frame the view, and note either of two distances: (1) the distance, v, from the viewframe to one eye — the other should be

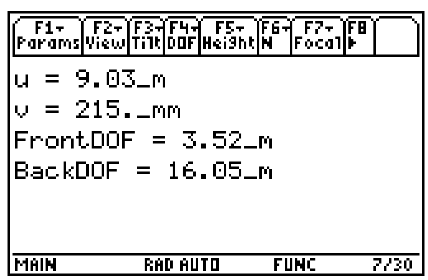
Figure 2: Gadgets

closed, since a camera has only one eye; or (2) the distance, u, to the point of focus. The following two programs will provide depth of field and other information.

3.1 Program Name: Viewv(), From v

The program asks for two parameters. The input is on the left, the output on the right.





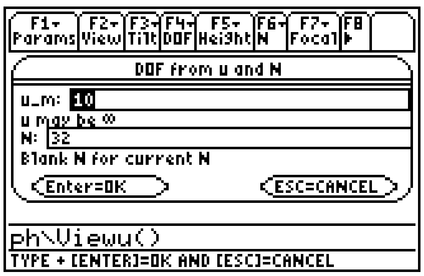
Here v is the distance in millimeters from the lens to the film plane. The f-stop, N, may be input since one sometimes needs to compare the effects of different N. If N is input as a blank, the current value of N, set by the Params() program, will be used. This current value is unchanged by Viewv().

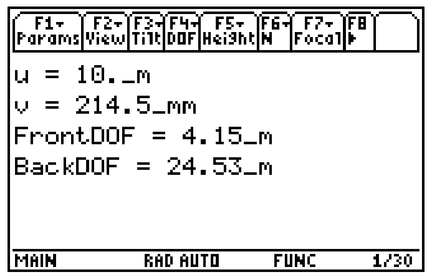
On the output, the u value is the distance from the lens to the in-focus subject. The FrontDOF and BackDOF values are the distances from u to the near and far DOF limits. In this illustration, the DOF is 19.57 m $(3.52+$

16.05).

3.2 Program Name: Viewu(), From u

The program asks for two parameters. The input is on the left, the output on the right.





The only difference from the previous program, is that u, the focus distance, is input. The f-stop, N, may be input since one sometimes needs to compare the effects of different N's. If N is input as a blank, the current value of N, set by the Params() program, will be used. This current value is unchanged by Viewu().

The u value is repeated from the input, and the v value calculated. The FrontDOF and BackDOF values are the distances from u of the near and far DOF limits.

3.3 Discussion

A viewframe improves the visualization of a composition. It aids in deciding the composition's orientation (landscape or portrait), and aids in the elimination of extraneous detail which the eye might otherwise not notice. There are other uses.

It may be used to find the appropriate focal length. Suppose a composition frames nicely in the viewframe when the subject is about 4 m away, and that the viewframe to eye distance, v, is about 220 mm. Any lens with less than 220 mm focal length may be chosen and then extended to 220 mm for focusing. A 210 mm might be appropriate. Of course one could also choose a 150 mm, but the extension to 220 m (150 mm + 70 mm) would result in a bellows factor of about 2. In this case, VIEWV using $v = 220$ m indicates the

FrontDOF and BackDOF values are 1.12 m and 2.17 m, assuming $F = 210$ and $N = 32$.

In making a decision about focal length, there is no need to be overly precise about v. A few millimeters either way will not change the composition much. The important thing is to decide on a focal length with approximately the coverage desired and leave the fine details to that time when the camera is focused.

Using a viewframe in this way is quite easy. What happens if one needs more depth of field, such as a foreground object? I raise the point, because some will attempt to adjust the DOF by changing their lens. There is nothing wrong with this if it meets the artistic needs, but often the resulting change distorts the composition. Extraneous detail which was outside the frame is now included, or the composition loosens up and becomes uninteresting. To keep the same composition when changing the lens, the camera must be moved. This will keep the magnification constant. Unfortunately, the DOF does not change very much when magnification is fixed. You might like to use the DOFMag() program which gives DOF as a function of magnification to establish the truth of this statement[4].

To increase DOF in a substantial way, the f-stop, N, must be changed. In this case doubling N from 32 to 64, produces $FrontDOF = 1.8$ m and $BackDOF = 8.21$ m. The composition remains unchanged.

4 Determining lens tilt

When the subject, lens, and film planes are parallel, focusing on any one part of the subject focuses on all parts. This is illustrated in Figure (1).

This is not true when the subject plane lies at an angle to the other planes. Focusing on one part of the tilted subject plane may cause other parts to be out of focus. To bring all parts of the subject into focus, the lens must be tilted so that all three planes meet at a line. This is Scheimpflug's rule. Figure (3) illustrates this rule. The parallel planes in Figure(1) also obey Scheimpflug's rule if one agrees, as is usual, that parallel planes meet at infinity.

[4]The magnification for the above illustration is about 0.021, which can be found from the Magv() program.

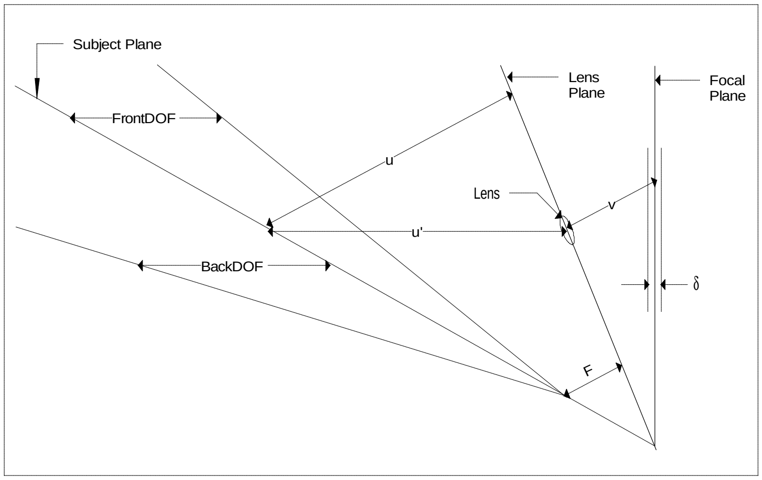


Figure 3: Tilted planes

If the subject plane is tilted, then it will meet the focal plane somewhere. The problem is to find the tilt angle of the lens such that a plane through it will also meet where the other two planes meet. Many experienced photographers decide on the tilt angle by cut-and-try mixed with considerable experience. If the ground glass were brightly illuminated, I would not find much fault with such a procedure: but it is not, it is almost always dim requiring a dark cloth to block light, and for some lens's with small maximum apertures, the ground glass can be grainy making nice judgments difficult. I prefer to calculate the angle from observations, and the following programs do this. The *BACK TILT* parameter which appears in the input menus of the following programs will be discussed in Section (4.4).

It is important to note, that although "tilt" is used in this section, the information applies equally to "swing."

4.1 TiltFrBk(), By back focusing

The easiest way to determine lens tilt is by focusing on two subjects which image near the top and bottom of the ground glass. Figure (4) shows this for two back positions, one when point A is in focus, and one when point C is in focus. The distance along the rail (in millimeters) between the focus points, together with the distance between the images (in millimeters) on the ground glass can be used to determine lens tilt. In figure (4) the Rail Δ is the distance from a to b, and the Glass Δ is the distance from b to c. The calculation of the lens tilt angle in degrees is given approximately by Wheeler's rule of 60. To wit $\theta \approx 60(b-a)/(c-b)$. The rule is not appropriate for close up work, but the TiltFrBk() program always gives the correct value.

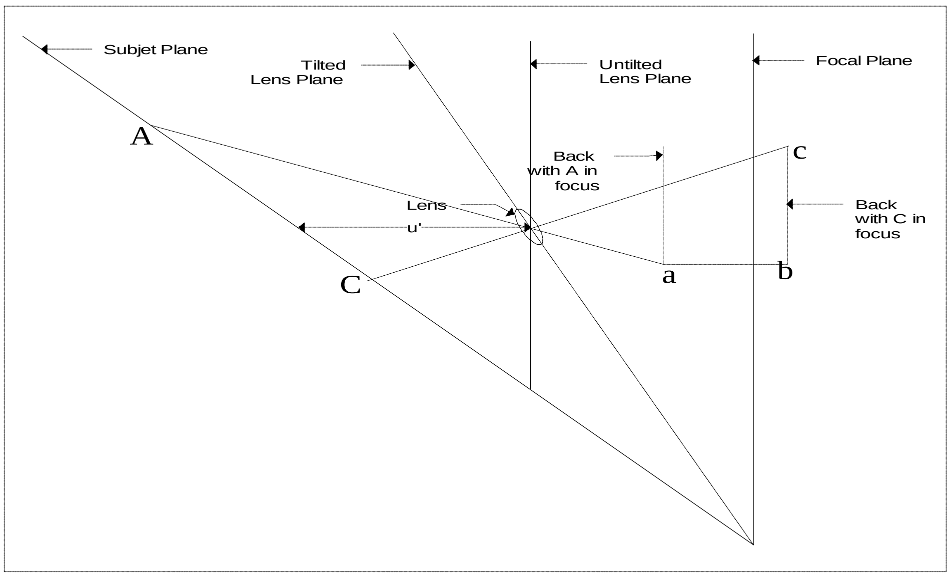
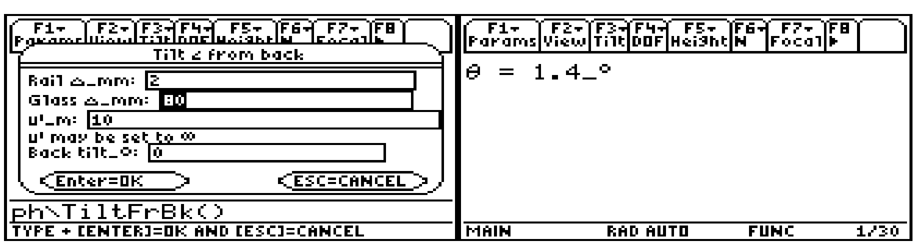


Figure 4: Tilt Diagram

The program input is on the left and the program output on the right. The input uses the distance u' shown in Figure(4) which is the horizontal distance to the subject plane. The tilt calculation does not make use of u', but it is used in calculating the slope γ of the subject plane, which parameter is saved for later use by DOFTilt().



The program outputs the lens tilt angle in degrees. The angle is positive for forward tilts, and negative for backward. Once the lens is tilted, and refocused, all points in the lens plane will be in focus.

In the case when the subject plane is vertical, the program should be informed of this by inputting the Rail Δ as 0 — a positive Glass Δ must always be input.

In refocusing after the lens tilt, those with center tilt cameras will find the focus point somewhere between the two previous points, while those with base tilt cameras will find it necessary to move the rear standard a considerable distance forward. Base tilt cameras move the lens in addition to rotating it, and the rear standard must be brought forward to adjust for this movement.

4.2 TiltFrDA(), Tilt from distance and angle

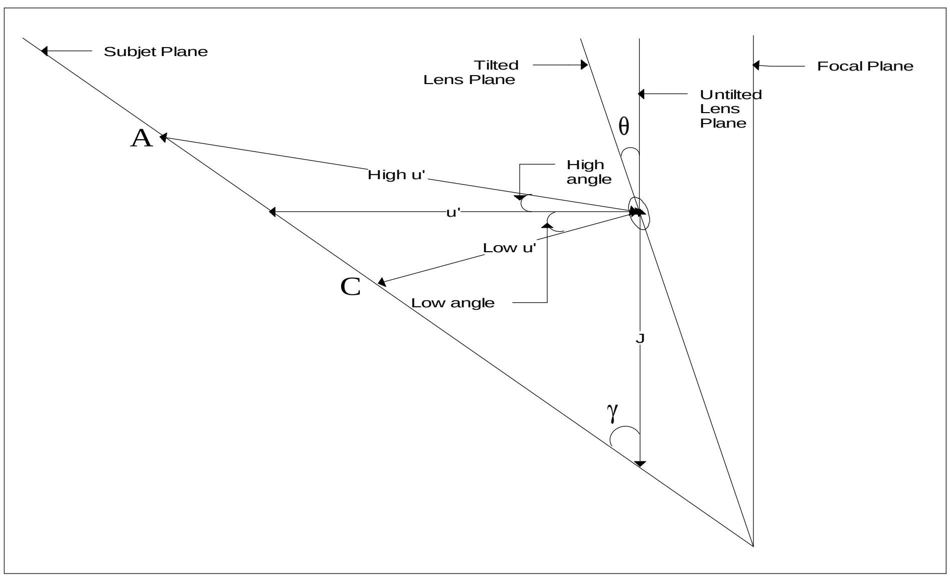
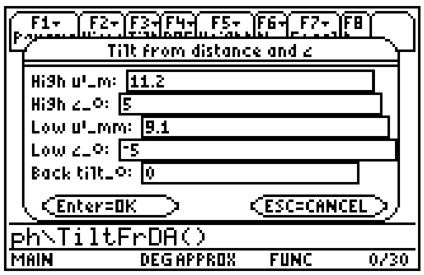
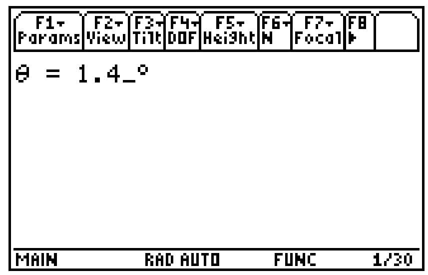


Figure 5: Distance and angle

Another way to determine lens tilt is by specifying the angles and distances to two objects in the desired subject plane. Figure (5) shows two points A and C on the subject plane. The distances and angles of these points are input to TiltFrDA(). The program assumes that angles above the horizon are positive, and negative below the horizon. One can buy palm size devices to measure angles above and below the horizon from shops that sell surveyors's equipment. Distance can be guessed. The calculations are not very sensitive to the far distance, but the near distance should be as accurate as possible. If a compass is available, then the program Tringlte() may be used to triangulate from two bearings — the compass must read in degrees or finer. The program input is on the left, the output on the right.



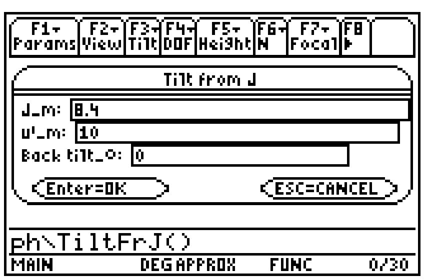


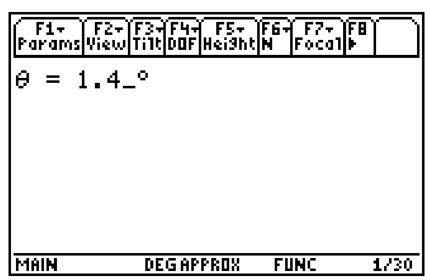
4.3 Tilt from geometric parameters

The next two programs ask for inputs derived from the geometry of the problem. These inputs can often be guessed with reasonable accuracy, and may prove easier to use in certain circumstances. Figure (5) shows the planes and rays involved for lens tilt calculations. The distance J is the vertical distance from the lens center to the subject plane, and γ is the angle that the subject plane makes with the lens plane.

4.3.1 TiltFrJ(), Tilt from J

The values of J and u' are guessed. The output is shown at the right.

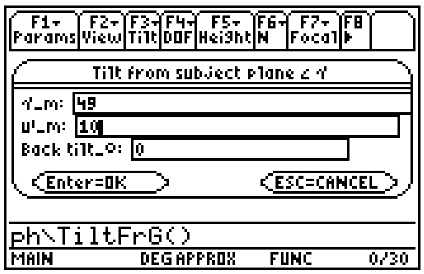


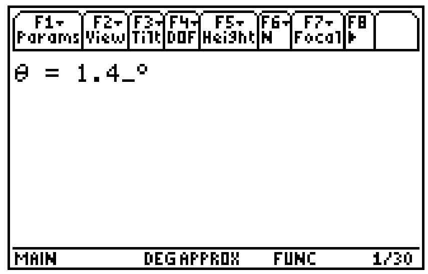


Merklinger (1992) and (1993) recommends this method using J alone, which is all that is required to calculate the tilt angle θ. TiltFrJ() asks for u' in order to calculate γ which is needed for DOFTilt().

4.3.2 TiltFrG(), Tilt from γ

The tilt angle is calculated from the subject plane angle γ and u'. The output is shown at the right.





4.4 Tilting the back

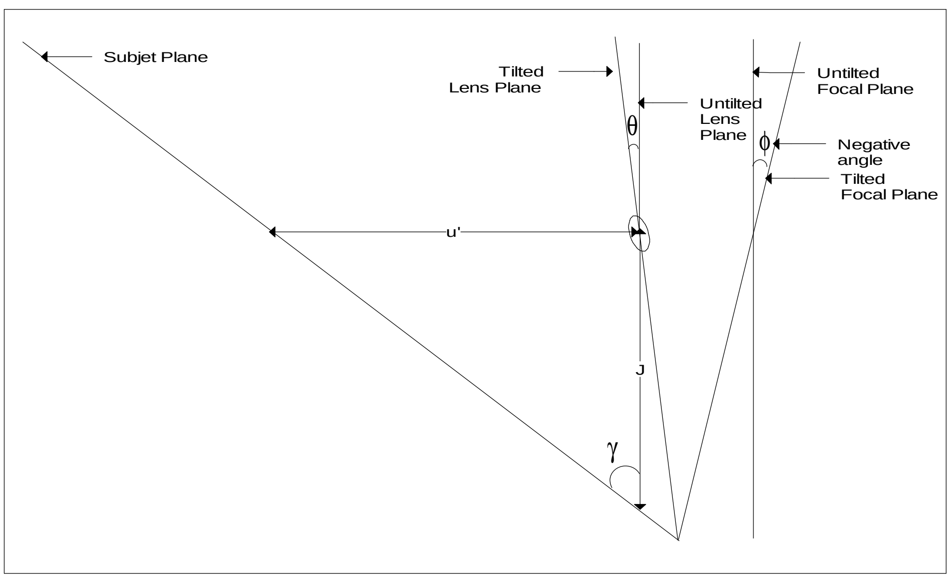
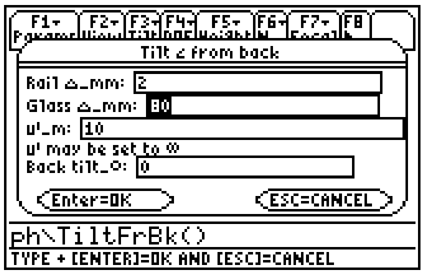
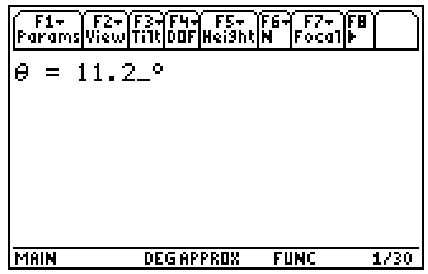


Figure 6: Tilted back

Tilting the back changes perspective by altering the way lines converge to vanishing points. Sometimes this change is desired. Scheimpflug's rule still applies when the back is tilted, as Figure (6) shows. The appropriate lens tilt angle will be calculated by each of the above programs when the

Back Tilt parameter is input. The input parameters should be obtained with an untilted back precisely as has been done above. The only change in the program input, is the setting of *Back Tilt* to a non-zero value. Consider the TiltFrBk() example above:





With the back tilted forward 10°, and the lens tilted 11.2° the subject plane will be in focus, although lines that were previously vertical will now diverge. It is assumed that positive back tilt angles imply a forward back tilt, and negative angles a backward tilt, just as they do for the lens tilt.

Back tilts can be used even when the subject plane is vertical, but the lens tilt is the same as the back tilt in this case – in other words the back and the lens axis remain parallel.

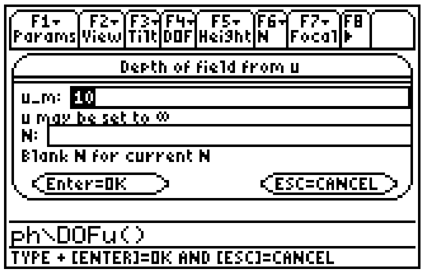
5 Determining DOF

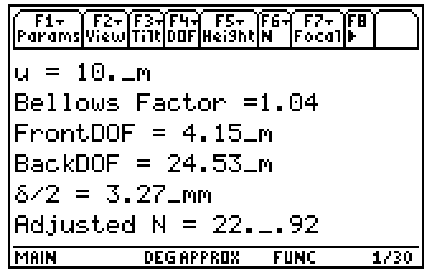
DOF is an important topic, and the photographer needs to assess it in several ways. The usual way is to calculate DOF as a function of u. But one can also use v, subject height, magnification, or the defocus δ. Programs are provided for each of these input values, as well as one that calculates the DOF along an arbitrary ray when the lens is tilted. Please refer to Figures (1) and (3). In addition, there is a program to calculate the hyperfocal distance, one that assesses the *blur* of objects at various distances, and two that translate DOF to and from δ.

All DOF programs report the same information, so this output will be described more fulsomely in the next subsection, (5.1) than in the other subsections.

5.1 DOFu(), As a function of u

The input is u. The output screen is shown on the right.





All DOF programs output u. For DOFu(), it is input. For other DOF programs it is calculated. The FrontDOF and BackDOF are distances about u. Thus the near point of the depth of field is $u - FrontDOF$, and the far point is $u + BackDOF$, as is illustrated in Figure (1).

The defocus distance δ is the distance on the rail corresponding to DOF, see Figure (1). For practical purposes, this distance is symmetrical about v, the distance on the rail cognate to the subject distance u. Thus, the points $v - \delta/2$ and $v + \delta/2$ are defocus limits, corresponding to near and far DOF limits. One can use the defocus limits to locate the near and far DOF planes. Simply move the standard back or forward by $\delta/2$ and observe those objects in sharpest focus — these correspond to objects on one of the DOF planes.

The adjusted f-stop, Adjusted N, shown is the bellows corrected f-stop. In this case, the original f-stop, N, was 32. Allowing for the bellows extension produces a value of 22_.92, which is not practically different from 32. The bellows factor by which the shutter speed may be adjusted is shown at the top of the screen.

> *Before releasing the shutter, it is always a good idea to perform a DOF calculation in order to check the bellows effect on the f-stop – surprises do occur.*

The DOF limits shown by the DOF programs are calculated assuming the input N. If the adjusted N differs practically from this N, and if the adjusted N is actually used to set the shutter, it will be necessary to recalculate the DOF limits by inputting the adjusted N to DOFu(). Of course a doubly adjusted N will be output, but this should be ignored.

5.2 DOFtilt(), From ray angle, for tilted lenses

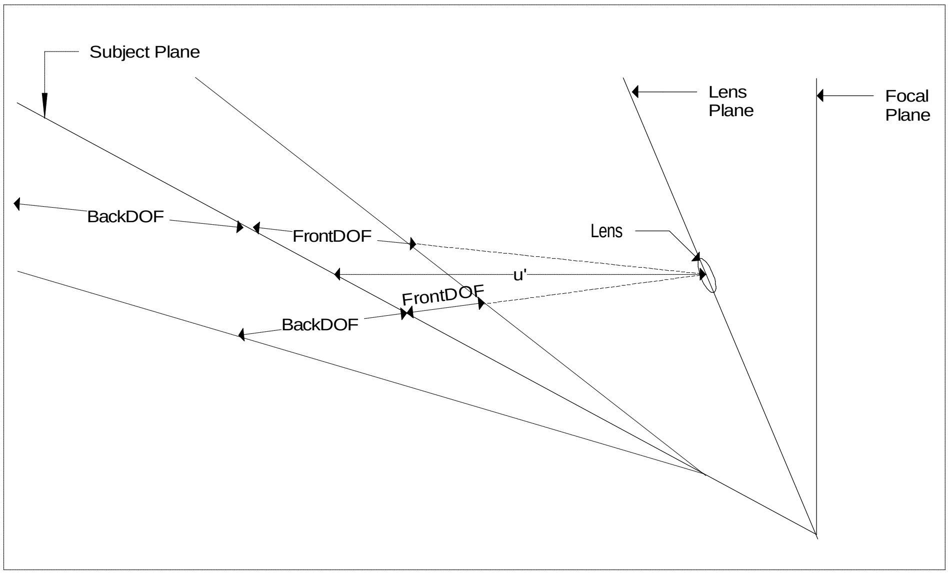
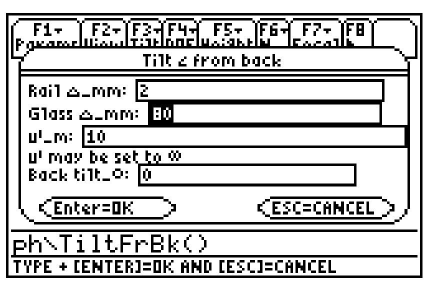


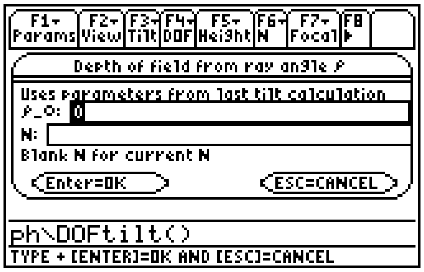
Figure 7: DOF for tilted lens

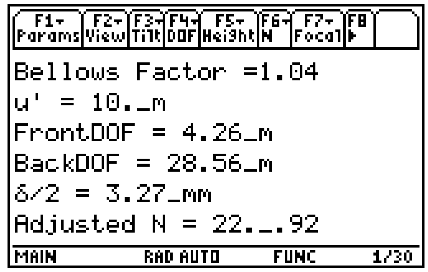
In order to set parameters that will be used by DOFtilt(), it is necessary to run one of the tilt programs. Other programs may intervene. DOFtilt() uses the parameters from the last run tilt program, and no other programs tinker with the saved tilt parameters.

The DOFtilt() program accepts a single input, an angle, and outputs the FrontDOF and BackDOF values along a ray at this angle to the horizontal. Figure (7) illustrates the situation. Assuming that TiltFrBk() has been run with the following parameters

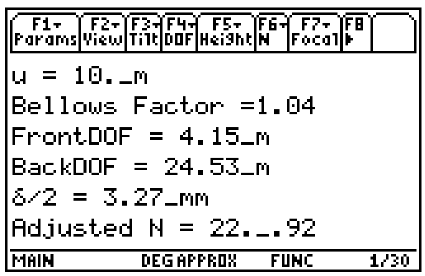


then inputting the angle zero into DOFtilt(), as on the left, produces the output on the right.





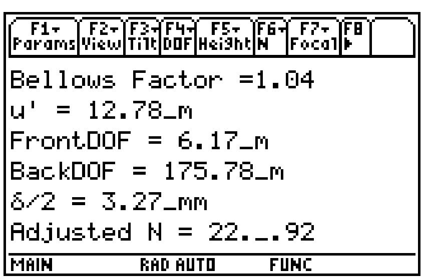
This calculation represents DOF for horizontal distances through a tilted lens. It can be compared with the output of DOFu() for an untilted lens:

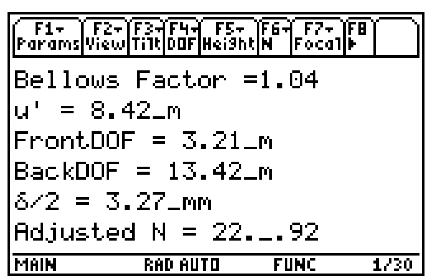


.

There seems to be little difference, but this is not the case when one looks along rays at an angle.

The output resulting from an angle of 10° is shown below on the left, and −10° on the right.



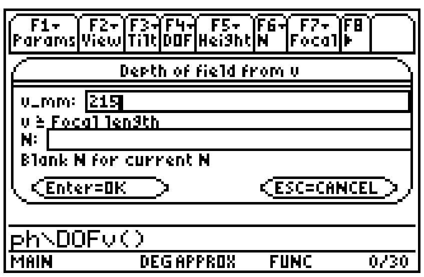


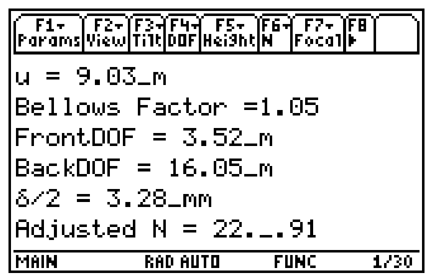
The effect of lens tilt on DOF is substantial.

It is important to note that the defocus, δ may be used to find the near and far DOF planes for tilted lenses, just as it can be for untilted lenses. The defocus limits are $v - \delta/2$ and $v + \delta/2$, and the distance v is the position of the rear standard when the subject plane is in focus. By moving the rear standard forward or back $\delta/2$ millimeters the DOF planes are those which are in sharpest focus.

5.3 DOFv(), From v

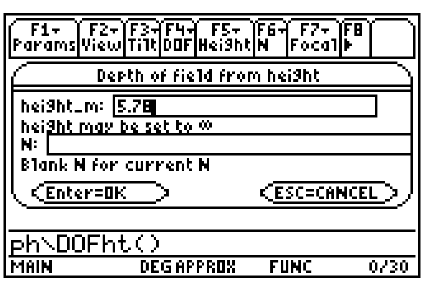
The distance between the lens and the film plane is v. It may be used to find DOF. Input and output for the DOFv() program are shown below. As a sidelight, note that the adjusted N value is little different from the nominal N of 32. This is because the extension $215 - 210 = 5$ mm is very small relative to $F = 210$ mm. For close up work, however, the adjusted N will be considerably different. Suppose one were interested in a 1 to 1 image, then the 210 mm lens would have to be focused at 440 mm, and the adjusted N would become 11_.83, and the DOF would shrink to about 10 mm.

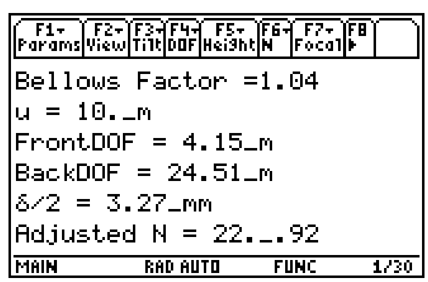




5.4 DOFht(), From subject height

By subject height is meant the vertical extent of the subject which will fill the long side[5] of the image on the film. In the present case with $u = 10$ m, the height is 5.78 m. The input and output are shown for this value.

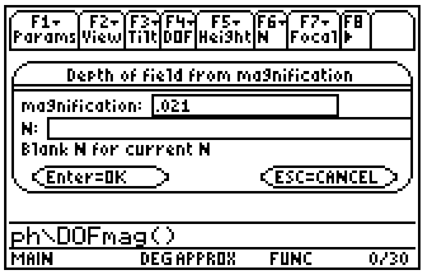


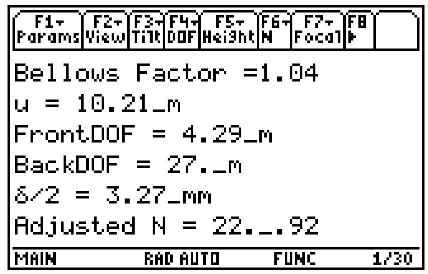


[5]The program chooses a film height of 124 mm for 4x5.

5.5 DOFMag(), From magnification

Magnification is the ratio v/u. In the present case it is $210/10000 = 0.021$ with both v and u in millimeters. Magnification is small for distant objects, but becomes large for close up photography. A magnification of 1 occurs when the image and subject heights are equal.

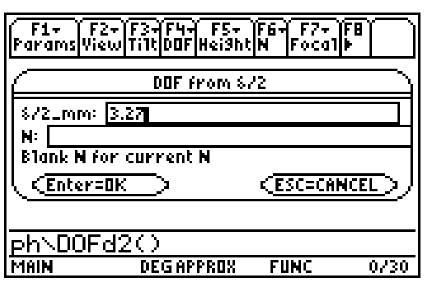


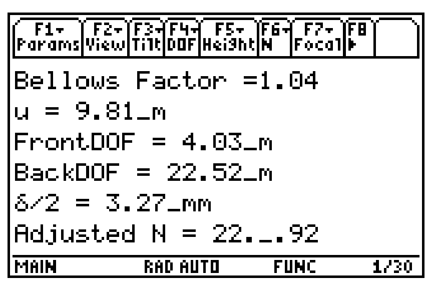


The output would have been slightly different had 0.02 been used for magnification. Whenever the output of two programs is unexpectedly different, look to the input values for an explanation. The calculations are done to many more places than are shown, and using too few places in the input can cause discrepancies.

5.6 DOFd2(), From defocus

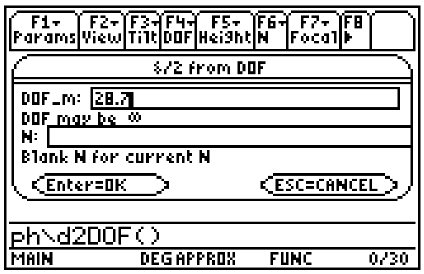
The defocus, δ, is the distance along the rail between the cognates of the near and far DOF: i.e. the difference between the locations of the two standards when the lens is focused on each of the two DOF limits. This is illustrated in Figure (1). For practical purposes, it is symmetrical about v, the position of the rear standard when the subject is in focus. Using $\delta/2 = 3.27$ mm to be consistent with the previous examples, the input and output screens appear as:

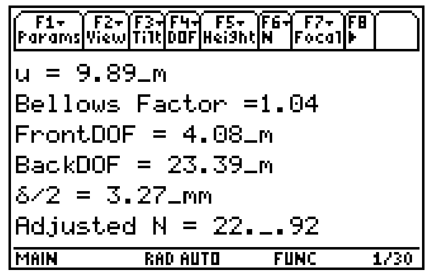




5.7 d2DOF(), From depth of field

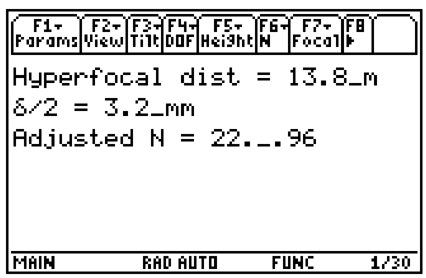
This program accepts the DOF. To be consistent with the previous examples, the DOF is set to 28.7 m.





5.8 Hyperfcl(), Hyperfocal distance

This simply returns the hyperfocal distance for the global parameters.



The $\delta/2$ and adjusted N are different from the previous programs because u has changed from 10 m to the hyperfocal distance of 13.8 m.

5.9 Fuzz(), Blurred images

Bokah is the Japanese term for out of focus or blurred objects. There is *good bokah* and *bad bokah*, but this is not the place to discuss its nature, and is only brought up here to introduce the fact that blurred images can form useful parts of a picture. Sometimes one has intrusive objects in the frame that need to be blurred, and sometimes it is just better to have a fuzzy area in a picture to support in a sense the main subject.

In any case, the Fuzz() output shows the degree of blurring for objects at specific distances. Fuzz() always calculates blurring values for the DOF

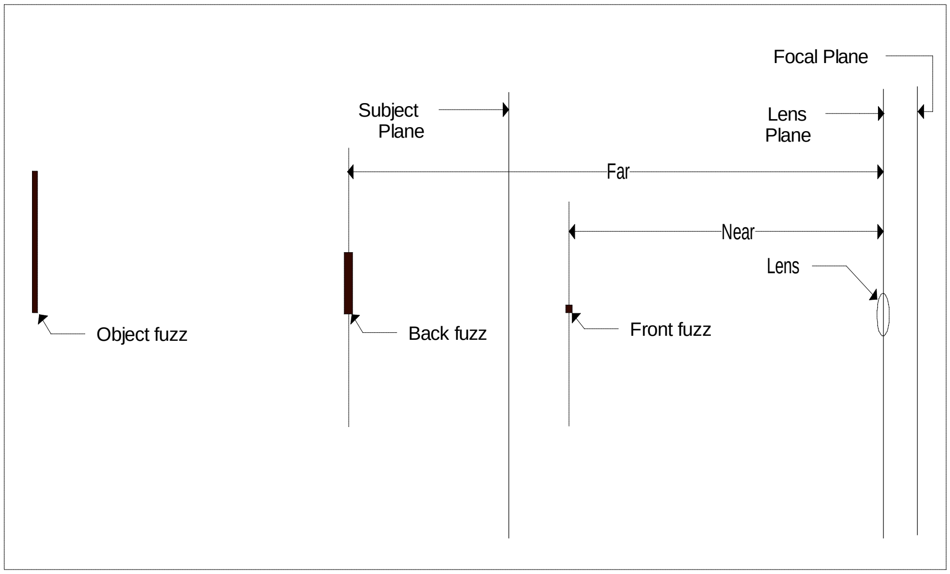
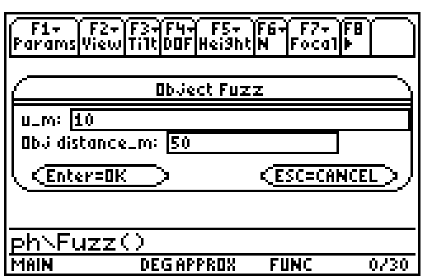
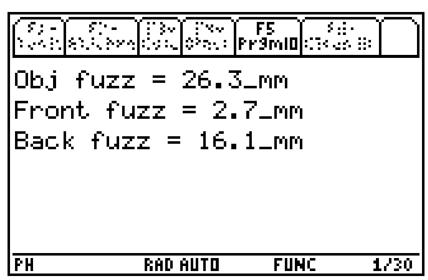


Figure 8: Fuzz at three distances

limits, and in addition it will calculate them for a user input distance. Consistent with the above examples, suppose $u = 10$ m and let us suppose we need to know about an object at 50 m. Figure (8) shows the size of objects at three distances which image on the film at exactly twice the diameter of the circle of confusion. The input and output are:



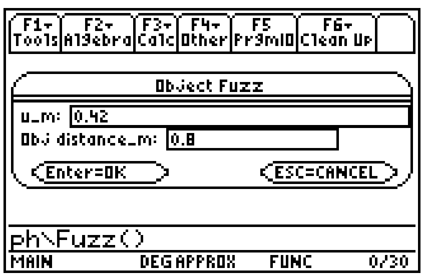


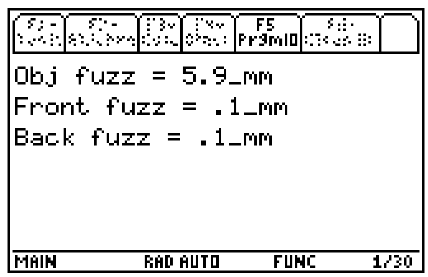
The *fuzz* values reported in the output, are the sizes of objects at the given distances which will image as twice the diameter of the circle of confusion[6]Thus

[6]To find the size of an object which images k times the diameter of the circle of confusion, one should multiply the size, s, output by the program, by $k - 1$. For $k = 2$ the multiplier $k-1$ is 1, and this is s as output by the program. The size of the object imaging

objects at the Near DOF of size 2.7 mm will barely be distinguishable. Similarly, objects of size 16.1 mm at the Far DOF limit will not be distinguishable. At 50m, a 26.3 mm object will not be distinguishable. This means that inch high lettering on an intrusive billboard some 50 m distant, will not be readable in a print. The lettering would have to be at four or more times this size to be readable, and even then would be very fuzzy.

This is especially useful for macro photography. Suppose you are photographing a flower at one-to-one magnification and there is an unavoidable object in the background. How visible will this object be? For a 210mm lens, one-to-one magnification puts the subject plane 420mm in front of the lens. Suppose the objectionable object is twice this far, say 800mm away from the lens. The Fuzz() input and output are:





If the object is smaller than 5.9 mm, then it will not be visible. If it is larger, say 25mm, then it will be very fuzzy because its top edge will not be distinguishable from a line 1/4 of this distance down. The size of the object will thus correspond to about four times the circle of confusion on the film, clearly a negligible amount.

6 Finding the subject's height

By subject height is meant the height of the subject that just fills the long side of the film image. The ratio of the long side of the film to the height is the magnification[7]. The height may be calculated from any of several parameters. Five programs are provided. They and their parameters are shown in Table (3).

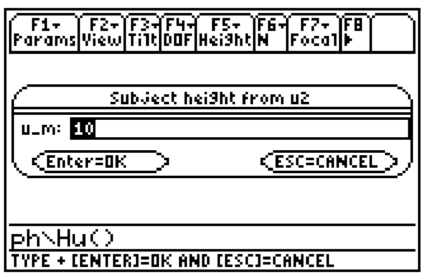
at four times the diameter of the circle of confusion would be $(k-1)s = 3s$. Using $4s$ instead, however, does little harm.

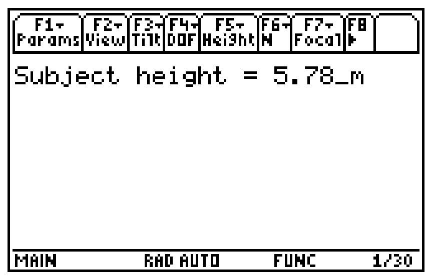
[7]Subject and film diagonals are often used instead of the height.

Program	Parameter
Hu()	u
Hv()	v
Hm()	magnification
HDOF()	FrontDOF
Hd2()	$\delta/2$

Table 3: Height program parameters

All input screens are similar, requiring a single parameter. Only the Hu() program will be illustrated. The output screen shows the subject height in meters. The input and output screens are:



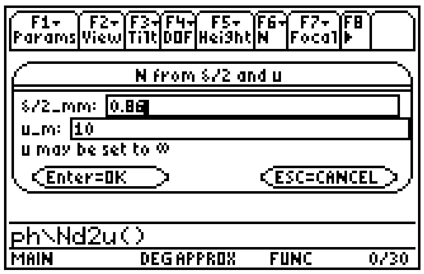


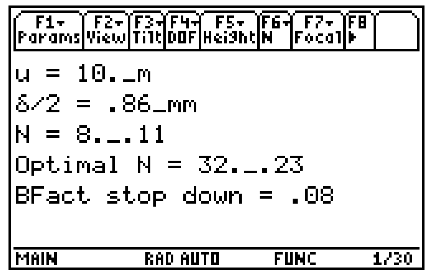
7 Finding the f-stop, N

There are three programs, NNearFar(), NDOFu() and Nd2u(), which calculate the f-stop, N.

7.1 Nd2u(), N from $\delta/2$ and u

The inputs are $\delta/2$ and u, as shown on the input screen at the left. The output is shown on the right.





The bellows correction shown at the bottom of the output depends only on the bellows extension and is thus the same for any N. It is given in stops, and may be subtracted from whatever N one selects. Thus 8_.11 becomes 8_.03 and 32_.23 becomes 32_.15.

Figure (9) illustrates the situation for the case when the lens is focused at infinity. In this figure, the optimum N line may be visualized as the crest of a mountain with the land sloping downward away from it. The "10 l/mm N" curve relates N to δ such that the on-film resolution is 10 l/mm. The "40 l/mm N" and "20 l/mm N" curves do the same for 40 l/mm and 20 l/mm respectively. The extent of the δ scale is appropriate for the 4x5 format. As a contrast, the dotted box at the lower left represents the δ scale appropriate for the 35 mm format. The reason that manufacturers choose $N = 16$ or $N = 22$ as the maximum f-stop for 35 mm should be clear from this figure.

Should one use N or optimum N? It all depends on what is wanted. The N output by the programs, corresponds to 10 l/mm, which is appropriate for an 8x10 print. The optimum N will of course support larger prints, but there seems little reason for choosing the optimum unless such large prints are the goal, and even then as may be seen in Figure (9) the optimum N will be less than 20 l/mm for δ's greater than about four. The program Resoltn() may be used to calculate the resolutions for particular combinations.

7.2 NNearFar(), N from near and far values

NNearFar() accepts Near and Far values for which it finds the N that will make them Near and Far DOF limits.

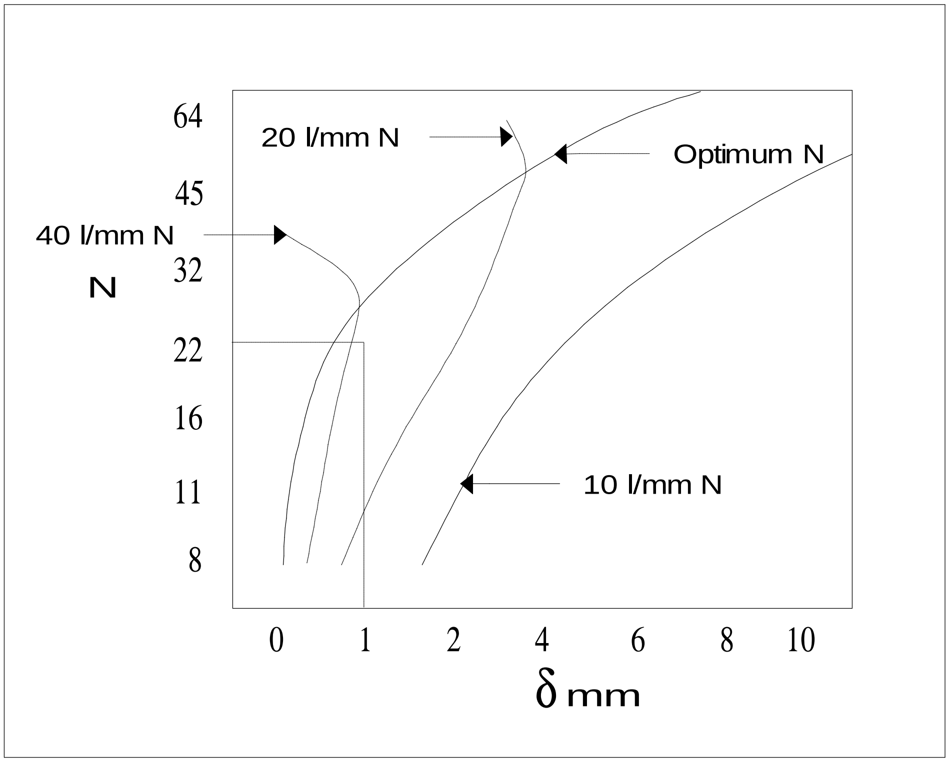
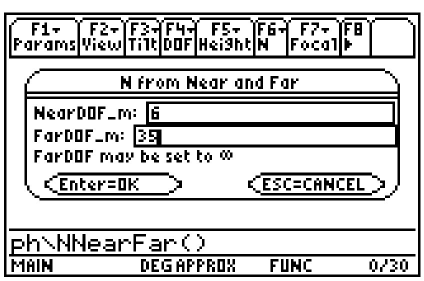
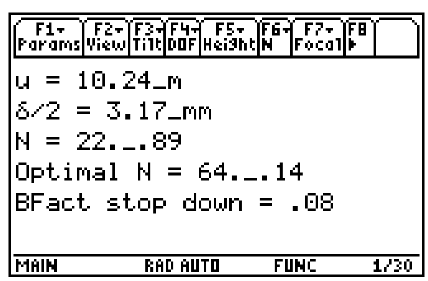


Figure 9: Optimum and resolution curves

The input and output screens for NNearFar() are as follows. Do not confuse the "Near" and "Far" values with "FrontDOF" and "BackDOF" values.

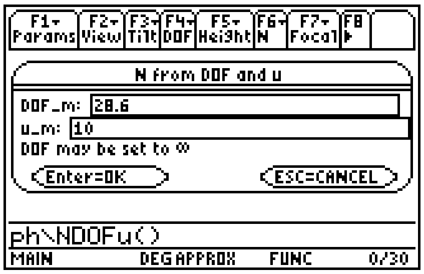


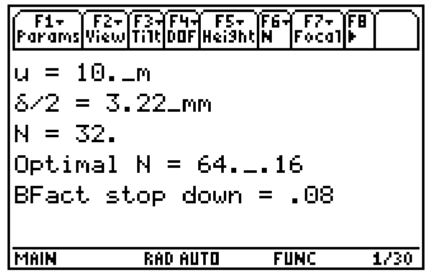


The calculated N is about 32. In addition the N which produces maximum resolution is shown as the optimal N.

7.3 NDOFu(), N from DOF and u

NDOFu() accepts DOF and u, and finds N. The input screen is on the left, the output on the right.





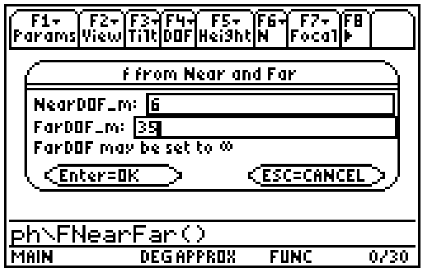
8 Finding the focal length

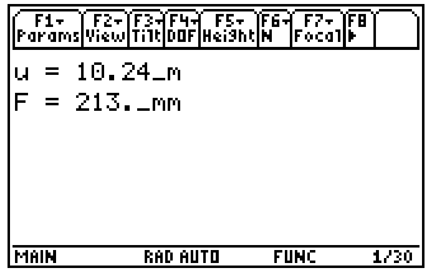
Focal length can be chosen in a variety of ways. The best way is to use a viewfinder to obtain a proper framing of the scene; however, other ways are possible. One may choose two distances, and map them into the DOF limits by a proper choice of focal length. Similarly, one may fix some other parameter, such as DOF or u, and calculate the focal length. Five programs are provided. They and their input parameters are show in Table (4).

Program	Parameters	
FNearFar()	Near	Far
Fd2u()	$\delta/2$	u
FDOF()	DOF	magnification
Fmu()	u	magnification
Fmv()	v	magnification

Table 4: Focal length programs

The program FNearFar() will be illustrated. The input and output screens are shown below. The F returned is the focal length required to make 6 m and 35 m the DOF limits.





9 Finding the magnification

Magnification is the ratio of the film size to the subject size[8], thus for a 2 m tall subject imaged on a 4x5 film with long side 124 mm, the magnification is 124/2000 = 0.062. Magnification controls the appearance of a picture, in that the subject size will fill the same area of the film if the magnification is constant. A picture taken with a 210 mm lens will frame the same composition as one taken with a 600 mm lens if the magnification for the two is the same. For example, in the case of a 2 m tall subject taken with a 210 mm lens on a 4x5 camera, the subject must be 3.6 m away. For a 600 mm lens, the subject must be 10.2 m away. The uvm() program may be used to confirm this.

Two pictures with the same magnification taken with different lenses may or may not appear identical – there can be a difference in resolution. The difference will be quite small, as may be seen by checking the DOFm() program which produces the values in Table (5). The f-stop in this table is $N = 32$ for both lenses.

Of course changing N will produce considerable differences in resolution, since DOF changes dramatically as N changes, but the point that has been made is that lens changes have little effect when magnification is constant.

There are four programs for magnification, as shown in Table (6).

[8]Any film dimension may be chosen. Each dimension will produce a slightly different value. The long side of the image is chosen here. Magnification is also equal to v/u.

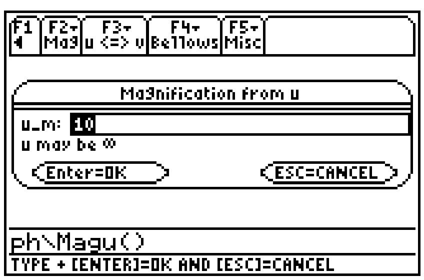
Lens	Distance	FrontDOF	BackDOF
210 mm	3.6 m	0.71 m	1.17 m
600 mm	10.6 m	0.81 m	0.97 m

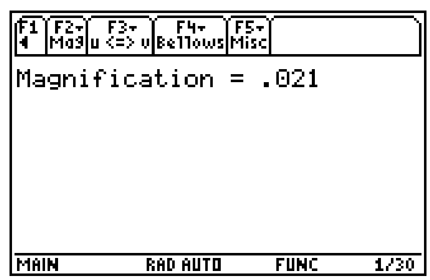
Table 5: Constant magnification for 2 m subject on a 4x5

Program	Parameter
Magu()	u
Magv()	v
Magh()	height
MAGd2()	$\delta/2$

Table 6: Magnification program parameters

The input and output for Magu() is shown below:





10 u <=> v

The parameters u and v satisfy an equation called the lens equation, and one may be computed from the other. In addition, either may be obtained from other parameters[9]. Four programs are given here. The utofromv()

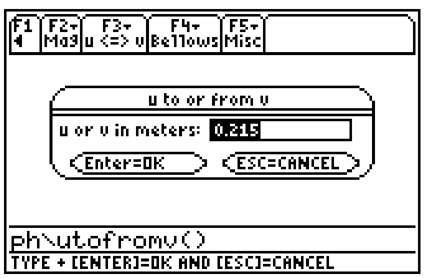
[9] Do not confuse u with u' which is used in connection with tilted lenses. Both represent distances from the lens to the subject, but only u for non-tilted lenses satisfies the lens equation. The lens equation may be used with tilted lenses, but u' is not the quantity needed.

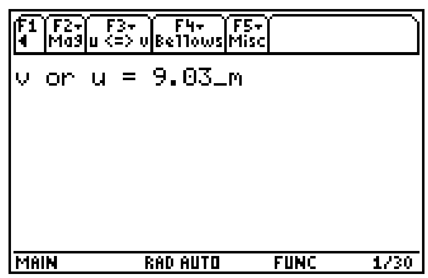
program translates u into v or v into u. The others produce u and v from magnification, subject height, and $\delta/2$. The programs and their parameters are given in Table (7).

Program	Parameter
utofromv()	u or v
uvh()	height
uvm()	magnification
uvd2()	$\delta/2$

Table 7: u and v program parameters

The input and output for utofromv() with v as input are shown below. Note that the input must always be in meters.





11 Bellows Factor

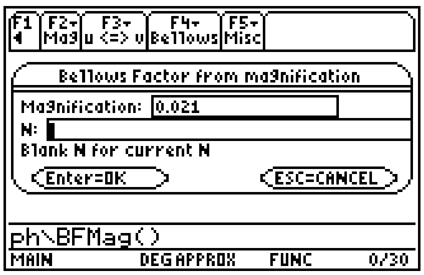
The f-stop that is read from a light meter strictly applies to the situation when the lens is focused at infinity. When the focus point is closer to the the lens, the lens must be extended and this decreases the amount of light reaching the film by the inverse square law. That is if the lens is moved twice as far out, the exposure will be one quarter of the original. To compensate for this, one should multiply the shutter speed by a factor. The factor is called the "bellows factor." The programs in this section give the bellows factor as a function either of the magnification or of the lens extension. They also translate the bellows factor into stops so that one may adjust the f-stop instead of the shutter speed if desired.

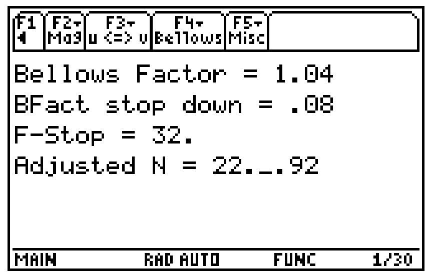
The programs and their parameters are given in Table (8).

Program	Parameter
BFMag()	magnification
BFext()	extension

Table 8: bellows factor program parameters

The input and output for the BFMag() program are shown below.



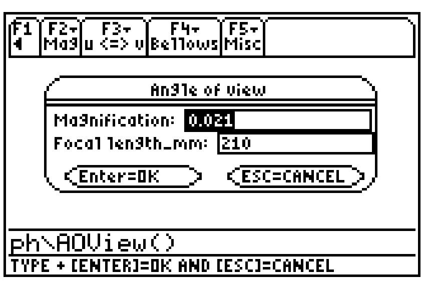


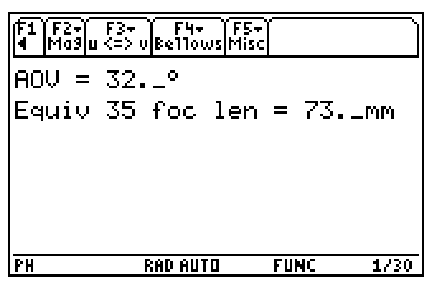
12 Miscellaneous programs

This section documents a number of miscellaneous programs such as AOView() for finding the angle of view, and StopMotn() for calculating the shutter speed required to stop motion.

12.1 AOView(), from magnification and focal length

The input and output are as shown.

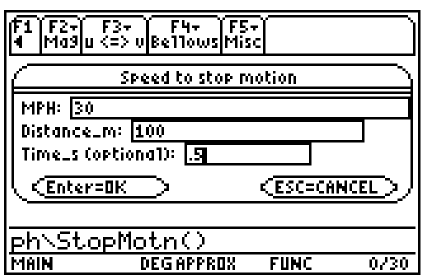


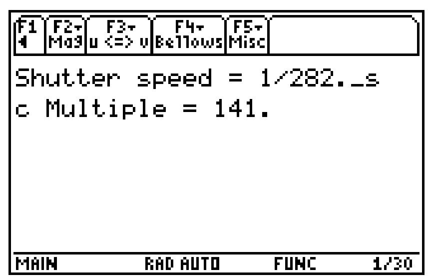


The last line of the output gives the focal length of a 35 mm lens with the angle of view shown.

12.2 StopMotn(), shutter speed to stop motion

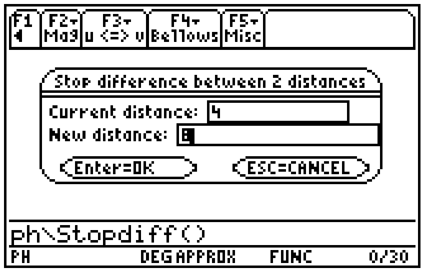
The screen on the left shows input for a 30 mph hour object 100 meters away. The direction of motion is assumed to be at right angles to the lens axis. The output on the right indicates that a shutter speed of 1/282 seconds will stop this motion. One may not always be able to shoot at the indicated speed, and some idea of the degree of blurring is useful. If the desired shutter speed, in seconds is input, then the degree of blurr is shown on the output screen. The output gives the distance the image of the subject moves in units of c, the diameter of the circle of confusion. A time of 1/2 second was input on the screen on the left and the output screen shows that the image of a subject will cover a distance of 141 times c during the 1/2 second exposure. It will be very blury. As a rule a subject that moves only two or three times c may be acceptable.

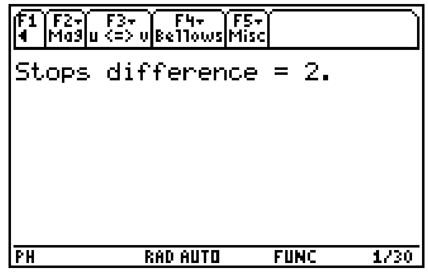




12.3 Stopdiff(), stop difference for light movement

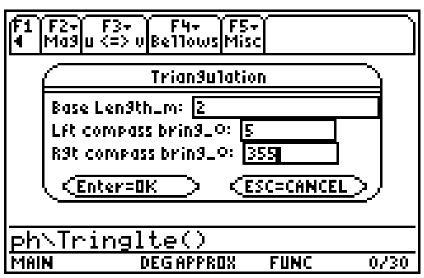
If the lights in a studio are moved, the f-stop will change. This program gives the stop difference required to make the adjustment. On the left, the lights are at 4 m and they are to be moved to 8 m. The output on the right shows that the f-stop must be increased by 2 stops.

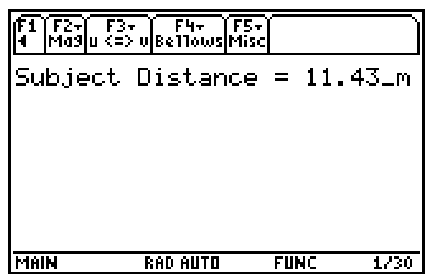




12.4 Tringlte(), triangulation

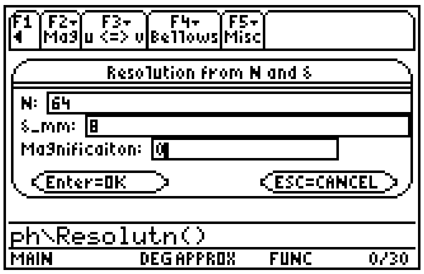
The distance to an object may be found by triangulation. The input screen on the left requires the base distance and two compass readings. I generally choose a two meter base, since I carry a retractable pocket measure and it is easy for me to hook the end on the camera and move out one meter on each side. My compass is shown among the gadgets in Figure (2).

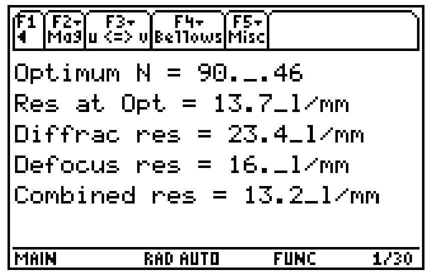




12.5 Resolutn(), resolution

This is a theoretical calculation of no practical use in the field, but included for completeness. With it one may calculate the resolutions shown in Figure (9). On the left one sees the input screen with δ entered as 8. Note the entry is not $\delta/2$ as with other programs. The output screen on the right shows the resolution at the optimum for $\delta = 8$. Below this are the resolutions from two different sources which combine to produce the final resolution. The first of these is the diffraction resolution, and the second is the defocus resolution at a $\delta/2$ distance from the plane of exact focus. The two resolutions are combined using root mean square on their inverses – this is strictly an empirical combination since there is no simple theoretical way to combine them. It is interesting to note that the resolution due to diffraction is actually the largest resolution shown, which should give pause to those who ascribe poor quality to diffraction – in this case, the principle cause is extreme defocus.





13 Technicalities

Parameters used by the programs are shown in Table (9): they are stored in directory z. They may be changed if desired, but remember that they will be reset when one of the indicated programs is run.

Parameter	Set by	Explanation
F	Params	Focal length
N	Params	f-stop
c	Params	diameter of the circle of confusion
Dm	Params	Film long side in mm
FS	Params	Film size 35,645,67,69,45,810
γ	Any tilt program	Slope of subject plane
J	Any tilt program	Vert distance from lens to subject plane
θ	Any tilt program	Lens tilt angle
α	Any tilt program	Back tilt angle

Table 9: Parameters

BIBLIOGRAPHY

1. Bond, H. (1998). *Photo Techniques*, May/June

2. Merklinger, Harold, M. (1992) *The ins and outs of focus.* Ottawa, Canada.

3. Merklinger, Harold, M. (1993) *Focusing the view camera.* Ottawa, Canada.

4. Jacobson, R.E.; Ray, S.F.; and Attridge, G.G. (1988). *The manual of photography.* Eighth Ed. Focal Press, Boston.

5. Ray, Sidney F. (1994) *Photographic optics.* Focal Press, Boston.

6. Sears, F.W. (1958) *Optics*, Addison Wesley, Reading MA.

7. Stroebel, Leslie; Compton, John; Current, Ira; and Zakia, Richard. (1986) *Photographic materials and processes.* Focal Press, Boston.

8. Stroebel, Leslie. (1993). *View camera technique.* Focal Press, Boston.

Directory	Program Name	Page	Description
	Params()	8	Set and display parameters
View			Viewframe information
	Viewv()	10	u,FrontDOF,BackDOF from v
	Viewu()	11	u,FrontDOF,BackDOF from u
Tilt			Calculates tilt angle
	TiltFrBk()	13	By back focusing
	TiltFrDA()	16	From distances and angles
	TiltFrJ()	17	From u and J
	TiltFrG()	18	From subject plane slope
DOF			Depth of field
	DOFu()	20	From u
	DOFTilt()	21	From ray angle, for tilted lens
	DOFv()	23	From v
	DOFht()	23	From height of the subject
	DOFmag()	24	From magnification
	DOFd2	24	From $\delta/2$
	d2DOF	25	$\delta/2$ from DOF
	Hyperfcl()	25	The hyperfocal distance
	Fuzz()	25	Blurred subject size
Height			Subject height
	Hu()	27	From u
	Hv()	27	From v
	Hm()	27	From magnification
	HDOF()	27	From DOF
	Hd2()	27	From $\delta/2$
N			Calculates F-Stops
	NNearFar()	29	From near and far values
	Nd2u()	29	From $\delta/2$ and u
	NDOFu()	31	From DOF and u
Focal			Focal length
	FNearFar()	31	From near and far DOF limits
	Fd2u()	31	From $\delta/2$ and u
	FDOF	31	From DOF and u
	Fmu()	31	From DOF and magnification
	Fmv()	31	From v and magnification
Mag			Magnification
	Magu()	32	From u
	Magv()	32	From v
	Magh()	32	From subject height
	Magd2()	32	From $\delta/2$
u <=> v			u and v
	utofromv()	33	u to v or v to u
	uvh()	33	u and v from subject height
	uvm()	33	u and v from magnification
	uvd2()	33	u and v from $\delta/2$
Bellows			Bellows factors
	BFm()	34	From magnification and N
	BFext()	34	From extension and N
Misc			Various utilities
	AOView()	35	Angle of view from magnification and focal length
	StopMotn()	36	Shutter spread to stop motion
	StopDiff()	36	Stop difference for two light distances
	Ttinglte()	37	Triangulates to find subject distance
	Resolutn()	37	Theoretical resolution