LV Lightvalues Exposure Calculator

Dietmar G. Schrausser orcid.org/0000-0002-4924-8280
Correspondence: dietmar.schrausser@gmail.com
Karl-Franzens University, Graz, Austria

Overview

Android application Lightvalues LV[1] for calculating *aperture values* Av corresponding to *light values*[2] by *shifting* the time value $Tv = s^{-1}$ or arithmetic ISO speed value S in steps k according to the common classification (s. Fig. 1), where

$$Tv_{n-k} = Tv_n \cdot 2^k, \quad S_{n+k} = S_n \cdot \sqrt{2}^k, \tag{1}$$

$$Tv_{n+k} = \frac{Tv_n}{2^k}, \quad S_{n-k} = \frac{S_n}{\sqrt{2}^k} \tag{2}$$

and

$$Av_{(Tv_n)} = Av_{(Tv_{n+k})} \cdot \sqrt{2}^k = \frac{Av_{(Tv_{n-k})}}{\sqrt{2}^k}, \tag{3}$$

$$Av_{(S_n)} = Av_{(S_{n-k})} \cdot \sqrt{2}^k = \frac{Av_{(S_{n+k})}}{\sqrt{2}^k}. \tag{4}$$

Therefore Av is calculated from Tv or S as

$$Av_{Tv} = Av_{Tv_0} \cdot a_{Tv}, \qquad Av_S = Av_{S_0} \cdot a_S \tag{5}$$

with

$$a_{Tv} = 2^{\frac{1}{2}\log 2 \frac{Tv_0}{Tv}} = \mathrm{e}^{\frac{1}{2}\frac{\log(Tv_0)}{Tv}}, \tag{6}$$

$$a_S = 2^{\frac{1}{2}\log 2 \frac{S}{S_0}} = \mathrm{e}^{\frac{1}{2}\frac{\log S}{S_0}}. \tag{7}$$

The shutter speed is set in the range between $Tv = 32000$ and 2 hours, $Tv = 0.000138$, aperture ranges from $Av = 0.5$ to $Av = 152$ and speed S is set to range between ISO 0.4 and ISO 102400. On aperture, shutter speed and exposure see e.g. Roberts (1995), Beaver (2018), Bernacki (2020) and Simon et al. (2022).

Logarithmic speed $S°$ (s. Allbright, 1991) is transformed from arithmetic speed S by

$$S° = 10 \cdot \log 10(S) + 1 = \frac{10 \cdot \log(S)}{\log(10)} + 1, \tag{8}$$

$$S = 10^{\frac{S°-1}{10}}. \tag{9}$$

The *exposure value* Ev is calculated from Tv and Av, where

$$Ev = \log 2 \frac{Av^2}{Tv^{-1}} = \frac{\log(Tv \cdot \mathrm{Av}^2)}{\log(2)}, Tv = \frac{2^{Ev}}{Av^2}, Av = \frac{\sqrt{2^{Ev} \cdot Tv}}{Tv}. \tag{10}$$

The total luminous flux or *illuminance* E_V in *lux* lx, where $lx = \frac{lm}{m^2}$ results from Ev and S by

$$E_V = 250 \cdot \frac{2^{Ev}}{S}. \tag{11}$$

For logarithmic functions in general see e.g. Marsden and Weinstein (1985), Howie (2001) and Sobot (2021).

Presets for time and aperture combinations at ISO 100/21° (8) are given (s. Tab. 1), with aperture values Av are rounded to one decimal place. Custom *time-aperture-ISO* combinations for *exposure values* Ev (10) or *illuminance* E_V (11) can be achieved by *shifting* Av itself (s. Fig. 1).

Table 1. Exposure presets for Tv, Av and Ev (10) at ISO 100/21° by condition *cnd*.

cnd	Tv	Av	Ev
Sun	125	11	14
Cloud	125	8	13
Overcast	60	5.6	11
Dawn	15	4	8
Indoors	15	2.8	7

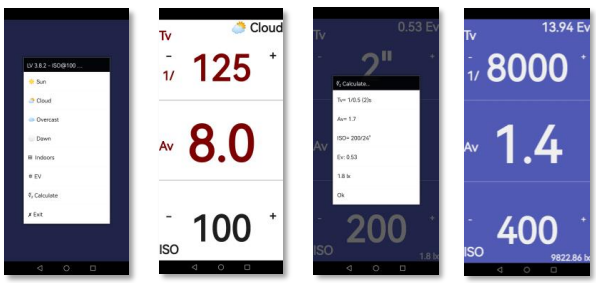
In addition, direct calculations (5) of aperture Av from shutter speed Tv (6) and S (7) can be performed (c.f. Schrausser, 2025). This should be used when shutter speeds outside the usual steps (1) (2) are present or when only *one* shutter speed is available, as in the case of the so-called mechanical *emergency* shutter speed (s. Tab. 2).

Table 2. Av for Tv (5) (6) at ISO 100/21° and ISO 400/27° with Ev (10) by condition *cnd*.

cnd	Tv				Ev
	250	*100*	*60*	*45*	
ISO 100/21°					
Sun	7.8	12.3	15.9	18.3	13.9
Cloud	5.7	9.0	11.6	13.3	13.0
Overcast	2.8	4.3	5.6	6.5	10.9
Dawn	1.0	1.6	2.0	2.3	7.9
Indoors	0.7	1.1	1.4	1.6	6.9
ISO 400/27°					
Sun	15.6	24.6	32.0	36.7	15.9
Cloud	11.4	18.0	23.2	26.6	15.0
Overcast	5.6	8.6	11.2	13.0	12.9
Dawn	2.0	3.1	4.0	4.6	9.9
Indoors	1.4	2.2	2.8	3.2	8.9

Further manuals or introductory literature on photography are given by e.g. Hedgecoe (1977, 2009) and Jacobson et al. (2000), see also Kenneth Mees (1931), Cannon and Hunt (1981), Hitchcock (1989), Current et al. (2000), Friedman and Ross (2003) or Pavlidis (2022).

Figure 1. Screenshots from LV Application.



Source

```
! ////////////////////////////////////////////////////////////
! // LV Lightvalues
! // Exposure calculator
! // by Dietmar G. Schrausser © 2025
! //
_name$="LV"
_ver$="3.8.2"
CONSOLE.TITLE _name$
INCLUDE strg.inc
INCLUDE lv.inc
GOSUB values
sw=-3
% // color switch  //
insw=1
% // input switch  //
tv=10
av=18
av1=18
```

[1] https://github.com/Schrausser/LV
[2] Light level for incident or reflected light on a logarithmic scale.

```
iso=18:iso$="100"
!
ml:
IF sw=1 :r=255:g=255:b=255:ENDIF
IF sw=0 :r=0  :g=0  :b=0  :ENDIF
IF sw=-1:r=80 :g=30 :b=30 :ENDIF
IF sw=-2:r=30 :g=80 :b=30 :ENDIF
IF sw=-3:r=80 :g=90 :b=180:ENDIF
GR.OPEN 255,r,g,b,0,1
GR.SCREEN sx,sy
dy=sy/3/4
txz0=sx/10
txz1=sx/2.5
txz2=sx/3.5
GR.TEXT.BOLD 1
IF insw=1 THEN GOSUB inpt
!
DO
 !
 insw=1
 GOSUB col
 !
% // AV shift      //
 GR.BOUNDED.TOUCH swavp,0,sy/3,sx/3,sy*2/3
 IF swavp=1
  IF av>1:av=av-1:rav=av:rtv=tv:ENDIF
  GOSUB sub1
 ENDIF
 GR.BOUNDED.TOUCH swavm,sx*2/3,sy/3,sx,sy*2/3
 IF swavm=1
  IF av<36:av=av+1:rav=av:rtv=tv:ENDIF
  GOSUB sub1
 ENDIF
 !
% // menue         //
 GR.BOUNDED.TOUCH swin,sx/3,sy/3,sx*2/3,sy*2/3
 IF swin=1
  GOSUB inpt:GR.CLS
 ENDIF
 !
% // TV shift      //
 GR.BOUNDED.TOUCH swtvp,0,0,sx/3,sy/3
 IF swtvp=1
  IF tv>1:tv=tv-1:av=av-2:ENDIF
  GOSUB sub1
 ENDIF
 GR.BOUNDED.TOUCH swtvm,sx*2/3,0,sx,sy/3
 IF swtvm=1
  IF tv<31:tv=tv+1:av=av+2:ENDIF
  GOSUB sub1
 ENDIF
 !
% // iso shift     //
 GR.BOUNDED.TOUCH swisop,sx*2/3,sy*2/3,sx,sy
 IF swisop=1
  IF iso<39:iso=iso+1
   av=av+1
% //              //
   GOSUB sub1
  ENDIF
 ENDIF
 GR.BOUNDED.TOUCH swisom,0,sy*2/3,sx/3,sy
 IF swisom=1
  IF iso>1:iso=iso-1
   av=av-1
   GOSUB sub1
  ENDIF
 ENDIF
 !
% // text TV       //
 iso$=iso$[iso]
 IF sw>-1
  IF tv=rtv THEN GR.COLOR 255,255/2,0,0,1
 ENDIF
 IF tv<1  THEN GR.TEXT.DRAW tx,sx/2,sy/3-dy,Tv$[1]
 IF tv>31 THEN GR.TEXT.DRAW tx,sx/2,sy/3-dy,Tv$[29]
 IF tv>0 & tv<32
  GR.TEXT.SIZE txz0
  GR.TEXT.ALIGN 1
  GR.TEXT.DRAW tx,0,dy,"Tv"
  IF tv<17
   GR.TEXT.DRAW tx,0,sy*3/12," 1/"
  ENDIF
  GR.TEXT.SIZE txz2
  GR.TEXT.ALIGN 2
  GR.TEXT.DRAW tx,sx/2,sy/3-dy,Tv$[tv]
 ENDIF
 GOSUB col
 !
% // text AV       //
 IF sw>-1
  IF rav=av THEN GR.COLOR 255,255/2,0,0,1
 ENDIF
 IF av<1  THEN GR.TEXT.DRAW tx,sx/2,sy*2/3-dy,Av$[1]
 IF av>36 THEN GR.TEXT.DRAW tx,sx/2,sy*2/3-dy,Av$[36]
 IF av>0 & av<37
  av1=av
  GR.TEXT.SIZE txz0
  GR.TEXT.ALIGN 1
  GR.TEXT.DRAW tx,0,sy/2+dy/2,"Av"
  GR.TEXT.SIZE txz1
  GR.TEXT.ALIGN 2
  GR.TEXT.DRAW tx,sx/2,sy*2/3-dy,Av$[av1]
 ENDIF
 GOSUB col
 !
% // text iso      //
 GR.TEXT.SIZE txz0
 GR.TEXT.ALIGN 1
 GR.TEXT.DRAW tx,0,sy-dy/3,"ISO"
 GR.TEXT.SIZE txz2
 GR.TEXT.ALIGN 2
 GR.TEXT.DRAW tx,sx/2,sy-dy,iso$
 !
% // Color switch //
 GR.BOUNDED.TOUCH t1,sx/3,sy*2/3,sx*2/3,sy
 IF t1
  insw=0
  sw=sw+1:IF sw=2 THEN sw=-3
  GR.CLOSE:GOTO ml
 ENDIF
 !
 GR.TEXT.SIZE txz0
 GR.TEXT.ALIGN 3
 IF dlg<>6
  GR.TEXT.DRAW tx,sx,sy/20,lv$[dlg]
 ELSE
  IF tv<31 & av1<35 & tv>1 & av>1 & iso>1 & iso<39
   GOSUB v0: GOSUB EV
  ENDIF
  GR.TEXT.DRAW tx,sx,sy/20,ev$+" Ev"
  GOSUB E_V
  GR.TEXT.SIZE sx/15
  GR.TEXT.DRAW tx,sx,sy-sy/100,e_v$+" lx"
 ENDIF
 GOSUB ln
 GR.RENDER
 !
UNTIL 0
!
ONERROR:
ONBACKKEY:
GOSUB fin
END
!
! // sub
////////////////////////////////////////////////////////////////
!
values:
ARRAY.LOAD tv$[], "+",
"32000","16000","8000","4000","2000","1000","500","250","125","6
0","30","15","8","4","2","1"+i$,"2"+i$,"4"+i$,"8"+i$,"15"+i$,"30
"+i$,"1'","2'","4'","8'","15'","30'","1h","2h","-"
ARRAY.LOAD av$[], "-
","0.5","0.6","0.7","0.8","1.0","1.2","1.4","1.7","2.0","2.4","2
.8","3.4","4.0","4.8","5.6","6.7","8.0","9.5","11","13.5","16","
19","22","27","32","38","44","54","64","76","88","108","128","15
2","+"
ARRAY.LOAD iso$[], "-
","0.4","0.6","0.8","1.1","1.5","2.2","3","4.4","6","8.8","12.5"
,"18","25","35","50","71","100","141","200","283","400","566","8
00","1131","1600","2263","3200","4526","6400","9051","12800","18
102","25600","36204","51200","72408","102400","+"
RETURN
!
col:
IF sw=1  THEN GR.COLOR 255,30 ,30 ,30 ,1
IF sw=0  THEN GR.COLOR 255,180,180,180,1
IF sw=-1 THEN GR.COLOR 255,255,30 ,30 ,1
IF sw=-2 THEN GR.COLOR 255,30 ,255,30 ,1
IF sw=-3 THEN GR.COLOR 255,240,240,240,1
RETURN
!
ln:
GR.LINE l1,0,sy*1/3,sx,sy*1/3
GR.LINE l2,0,sy*2/3,sx,sy*2/3
GR.TEXT.ALIGN 1
GR.TEXT.DRAW tx,0,sy*1/6,"  -"
GR.TEXT.DRAW tx,0,sy*5/6,"  -"
GR.TEXT.ALIGN 3
GR.TEXT.DRAW tx,sx,sy*1/6,"+  "
GR.TEXT.DRAW tx,sx,sy*5/6,"+  "
RETURN
!
sub1:
PAUSE 100:GR.CLS
RETURN
!
inpt:
ARRAY.LOAD lv$[],m1$+" Sun",m2$+" Cloud",m3$+" Overcast",m4$+"
Dawn",m5$+"  Indoors",m6$+" EV",m7$+" Calculate",_ex$+" Exit"
DIALOG.SELECT dlg,lv$[],_name$+" "+_ver$+" - ISO@100 …"
IF dlg=5:tv=13:av=12:iso=18:ENDIF
IF dlg=4:tv=13:av=14:iso=18:ENDIF
IF dlg=3:tv=11:av=16:iso=18:ENDIF
IF dlg=2:tv=10:av=18:iso=18:ENDIF
IF dlg=1:tv=10:av=20:iso=18:ENDIF
IF dlg=7
 IF tv<31 & av1<35 & tv>1 & av>1 & iso>1 & iso<39
  GOSUB v0
  GOSUB calc
 ENDIF
ENDIF
IF dlg=8:GOSUB fin:END:ENDIF
rtv=tv
rav=av
RETURN
!
calc:
GOSUB EV:GOSUB E_V
ARRAY.LOAD calc$[],"Tv= 1/"+tv0$+" ("+tv1$+")s","Av=
"+av0$,"ISO= "+iso0$+"/"+din$+"°","Ev: "+ev$,e_v$+" lx","Ok"
DIALOG.SELECT dlg2,calc$[],m7$+" Calculate…"
IF dlg2=1
 INPUT "Tv=…",tv01,VAL(tv0$)
 GOSUB AvTv
 GOTO calc
ENDIF
IF dlg2=2
 INPUT "Av=...",av01,VAL(av0$)
 av0$=STR$(av01)
 tv01=VAL(tv0$)
 GOSUB AvTv
 GOTO calc
ENDIF
IF dlg2=3
 INPUT "ISO=…",iso01,VAL(iso0$)
 GOSUB Aviso
 GOTO calc
ENDIF
IF dlg2=6 THEN GOTO inpt
RETURN
!
v0:
% // conversions //
```

```
IF tv<17
% // Tv < 1sec   //
 tv0$=Tv$[tv]
 tv1$=STR$(ROUND(1/VAL(tv0$),5))
ENDIF
IF tv>16
% // Tv >= 1sec  //
 SW.BEGIN tv
  SW.CASE 17:tv0$="1"      :tv1$="1"   :SW.BREAK
  SW.CASE 18:tv0$="0.5"    :tv1$="2"   :SW.BREAK
  SW.CASE 19:tv0$="0.25"   :tv1$="4"   :SW.BREAK
  SW.CASE 20:tv0$="0.125"  :tv1$="8"   :SW.BREAK
  SW.CASE 21:tv0$="0.0667" :tv1$="15"  :SW.BREAK
  SW.CASE 22:tv0$="0.0333" :tv1$="30"  :SW.BREAK
  SW.CASE 23:tv0$="0.0167" :tv1$="60"  :SW.BREAK
  SW.CASE 24:tv0$="0.0083" :tv1$="120" :SW.BREAK
  SW.CASE 25:tv0$="0.0042" :tv1$="240" :SW.BREAK
  SW.CASE 26:tv0$="0.0021" :tv1$="480" :SW.BREAK
  SW.CASE 27:tv0$="0.0011" :tv1$="900" :SW.BREAK
  SW.CASE 28:tv0$="0.00056":tv1$="1800":SW.BREAK
  SW.CASE 29:tv0$="0.00028":tv1$="3600":SW.BREAK
  SW.CASE 30:tv0$="0.00014":tv1$="7200":SW.BREAK
 SW.END
ENDIF
av0$=Av$[av1]
iso0$=iso$
din$=INT$(INT(10*LOG10(VAL(iso0$))+1))
% // DIN          //
RETURN
!
AvTv:
% // calc Av Tv   //
av0$=STR$(VAL(av0$)*EXP((0.5*LOG(VAL(tv0$)/tv01))) )
av0$=STR$(ROUND(VAL(av0$),2))
tv0$=STR$(tv01)
tv1$=STR$(ROUND(1/tv01,5))
RETURN
!
Aviso:
% // calc Av iso  //
av0$=STR$(VAL(av0$)*EXP((0.5*LOG(iso01/VAL(iso0$)))) )
av0$=STR$(ROUND(VAL(av0$),2))
iso0$=STR$(iso01)
din$=STR$(ROUND(10*LOG10(iso01)+1,1))
RETURN
!
EV:
% // calc Ev      //
ev$=STR$(ROUND(1/LOG(2)*LOG((VAL(av0$))^2/(VAL(tv0$))^-1),2))
RETURN
!
E_V:
% // calc E_V     //
e_v$=STR$(ROUND(250*(2^VAL(ev$)/VAL(iso0$)),2))
RETURN
!
fin:
PRINT _name$+" Lightvalues "+_ver$
PRINT "Copyright "+_cr$+" 2025 by Dietmar Gerald Schrausser"
PRINT "https://github.com/Schrausser/LV"
PRINT "DOI:10.5281/zenodo.16502602"
RETURN
! // END //
! //
```

References

Allbright, G. S. (1991). Emulsion Speed Rating Systems. *The Journal of Photographic Science, 39*(2), 95–99. https://doi.org/ 10.1080/00223638.1991.11737126

Beaver, J. (2018). Shutter Speed and Aperture. In *The Physics and Art of Photography*, 2, 3-1 to 3-6, 2053-2571. Morgan & Claypool Publishers. https://doi.org/ 10.1088/20 53-2571/aae504ch3

Bernacki, J. (2020). Automatic Exposure Algorithms for Digital Photography. *Multimedia Tools and Applications, 79*(19), 12751–76. https://doi.org/10.1007/ s11042-019-08318-1

Cannon, T. M., & Hunt, B. R. (1981). Image Processing by Computer. *Scientific American, 245*(4), 214–25. http://www.jstor. org/stable/24964586

Current, I., Compton, J. C., & Zakia, R. D. (2000). *Basic Photographic Materials and Processes*. Amsterdam: Elsevier Science; Technology. https://books.google. com/books?id=maKozwEACAAJ

Friedman, A., & Ross, D. S. (2003). *Mathematical Models in Photographic Science*. Berlin, Heidelberg: Springer. https://doi.org/10. 1007/978-3-642-55755-2

Hedgecoe, J. (1977). *The Photographer's Handbook: A Complete Reference Manual of Techniques, Procedures, Equipment and Style*. 1st ed. New York: Knopf. https://books.google.com/books?id=eyXrAAAAMAAJ

Hedgecoe, J. (2009). *New Manual of Photography*. London: Dorling Kindersley Limited. https://books.google.com/books?id=9N4C0HMzZFMC

Hitchcock, M. (1989). Field Photography; a Guide to Basic Equipment. *Journal of Museum Ethnography*, 1, 4–6. http://www.jstor. org/stable/40793474

Howie, J. M. (2001). The Logarithmic and Exponential Functions. In *Real Analysis*, 165–79. London: Springer. https://doi. org/10.1007/978-1-4471-0341-7_6

Jacobson, R., Ray, S., Attridge, G. G., & Axford, N. (2000). *Manual of Photography*. 9th ed. Oxfordshire, UK: Routledge. https:// doi.org/10.4324/9780080510965

Kenneth Mees, C. E. (1931). The Science of Photography. *Sigma Xi Quarterly, 19*(1), 1–19. http://www.jstor.org/stable/27824446

Marsden, J., & Weinstein, A. (1985). Exponentials and Logarithms. In *Calculus i*, 307–35. New York, NY: Springer. https://doi. org/10.1007/978-1-4612-5024-1_9

Pavlidis, G. (2022). *Foundations of Photography: A Treatise on the Technical Aspects of Digital Photography*. Cham: Springer International Publishing. https://doi. org/10.1007/978-3-031-06252-0

Roberts, G. (1995). Exposure. In *Mastering Photography*, 76–87. London: Macmillan Education UK. https://doi.org/10.1007/ 978-1-349-13506-6_5

Schrausser, D. G. (2025). HP_Prime_MATH: Manual. *Zenodo. June 2025*. https://doi. org/10.5281/zenodo.15713317

Simon, G., Vakulya, G., & Rátosi, M. (2022). The Way to Modern Shutter Speed Measurement Methods: A Historical Overview. *Sensors, 22*(5), 1871. https://doi. org/10.3390/s22051871

Sobot, R. (2021). Exponential and Logarithmic Functions. In *Engineering Mathematics by Example*, 51–66. Cham: Springer International Publishing. https://doi. org/10.1007/978-3-030-79545-0_4