

東広島天文台新スカイモニターの開発と遠隔操作 に向けた観測環境情報の統合化

永安 真弓

広島大学理学部物理科学科

B062478

高エネルギー宇宙・可視赤外線天文学研究室

主査 川端弘治 副査 生天目博文

2011年02月10日

概要

地上からの天体観測においては、大気の状態、雲の有無が大きな影響を及ぼす。また、天候の変化による観測中止・再開の判断の遅れが、観測時間を無駄にしたり、機器にダメージを与えることになる。よって、リアルタイムにこれらの観測情報を把握することが重要である。東広島天文台の観測環境を確認するものとして、湿度、風速等を調べる気象センサーと雨を感知する雨滴センサー、主に雲を判定するスカイモニターがある。私は、雲の視認性の向上と、予測不能な突発的天体の検知を目指し、全天を監視するスカイモニターのアップグレードを行った。また、観測環境情報を統合化して表示するシステムを構築した。

これまでのスカイモニターは、ノイズが大きく、また露出時間などの設定ができないため、昼間や薄明時、満月の夜は、サチュレーションを起こし視認性が悪かった。そこで私は、空間分解能が高く露出パラメータを遠隔で設定変更可能なカメラを導入し、空の明るさに合せた露出パラメータを計算し、撮影後画像の加工を行いWeb上で公開するソフトウェアの開発を行った。露出パラメータは、日の出・日没時刻との差をもとに、自前の計算式を用意し、数日間の試験観測によって調整した計算式を用いている（右図は薄明中で且満月時の画像取得例、雲と木星の確認が可能）。さらにスカイモニターに加え、他の観測環境情報を集約し、統合的に表示して、異常時には注意を促すシステムを構築した。



目次

第1章	序論	5
1.1	かなた望遠鏡	5
1.2	観測環境情報	6
1.3	スカイモニターとその要求仕様	7
1.4	既存のスカイモニターと改善点	8
1.5	本研究の目的	10
第2章	ハードウェア	12
2.1	カメラ本体及び魚眼レンズ	12
2.2	ハウジングと周辺機器	14
2.3	カメラの設置及びPCとの接続	17
第3章	ソフトウェア	20
3.1	ソフトウェアへの要求仕様	20
3.2	C言語プログラムによる自動画像取得プログラムの作成	20
3.3	最適露出パラメーターの計算法	24
第4章	性能評価と考察	30
4.1	試験観測	30
4.2	薄明時の露出パラメーター	30
4.3	雲認知度合の評価	35
4.4	月のフェーズによる雲の見栄え・恒星の限界等級	38
4.4.1	新月期 (月齢 0~3、26~29)	38
4.4.2	半月期 (月齢 4~10、19~25)	40
4.4.3	満月期 (月齢 11~18)	42
第5章	まとめと今後	44
付録A	ソースコード	46
付録A.1	画像の自動取得・加工・更新プログラム	46
付録A.2	露出時間計算プログラム	51
付録A.3	F値計算プログラム	57
付録A.4	ISO感度計算プログラム	60

目 次

1.1	かなた望遠鏡	5
1.2	かなた望遠鏡と可視赤外線同時撮像分光装置 TRISPEC で観測した GRB 080506 の可視 V バンドの画像。左から順に γ 線で光出してからの経過時間がそれぞれ 210 秒後、989 秒後、3289 秒後の画像であり黄色矢印で示した天体が次第に暗くなっていくのがわかる。	6
1.3	かなた望遠鏡による観測までの流れ。環境センサーは左上の東広島天文台の写真で黄色の丸で囲まれた部分に設置されており、そこで得られた情報を web サイト上に集約して公開し、それを確認して観測を行う。	7
1.4	既存スカイモニターの魚眼レンズ	8
1.5	既存スカイモニターのカメラ (WAT-129NCCD カメラ)	8
1.6	既存のスカイモニターで撮影した画像 (2009 年 3 月 14 日)。画像処理をしない状態で確認できる限界等級は 3 等星で、色レベルの調整をした後では約 5 等星までが視認できた。	10
2.1	新スカイモニターの魚眼レンズ (4.5mm F2.8 EX DC Circular Fisheye HSM)	12
2.2	新スカイモニターのデジタルカメラ (NikonD3000)	12
2.3	気象用密封ケース	15
2.4	OMRON 温度制御用デジタルサーモ	15
2.5	坂口電熱シリコンラバーヒーター	15
2.6	DC ファンモータ	15
2.7	12V 電源 EWS50-12	16
2.8	既存スカイモニターの装置図	16
2.9	新スカイモニターの装置図	17
2.10	既存スカイモニターのアルミ基盤と光学台の合成図	18
2.11	デジタルカメラに交換した後の筐体内部	19
3.1	スカイモニター (手前) と西にあるかなた望遠鏡のドーム (奥)	21
3.2	加工前の画像。2011 年 1 月 27 日 16:30 に撮影した画像の生データで、東西南北が一致しておらず、左上にかなた望遠鏡のドームとその横に夕日が見える。また、左右の黒い部分は不要。	21

3.3	加工後の画像 (2011年1月27日 16:30)。方角が正しく調整され、余分な部分が切り取られている。左下には撮影時刻、右下には露出パラメーターを書き込んだ。	22
3.4	加工後の夜の画像 (2011年1月26日 21:30)。「N」の下の黄色い丸は北極星で、下の方にはオリオン座も確認できる。	23
3.5	F 値設定のモデル曲線 (初期) のグラフ。縦軸は F 値、横軸は時間 (分) を表す。横軸の目盛は 150 分間である。赤線は計算による理想の値、青線は実際に設定する値。	26
3.6	ISO 感度設定のモデル曲線 (初期) のグラフ。縦軸は ISO 感度、横軸は時間 (分)、1 目盛は 150 分を示す。赤線は計算による理想の値、青線は実際に設定する値。	27
3.7	露出時間設定のモデル曲線 (初期) のグラフ。縦軸は露出時間、横軸は時間 (分)、1 目盛は 150 分を示す。赤線は計算による理想の値、青線は実際に設定する値。縦軸には露出時間の対数 (log) を使用した。	27
3.8	露出時間のリニアと対数のグラフ。青線はリニアのままのグラフで、赤線は対数をとったグラフである。日の出、日没前後で違いが大きい。	28
4.1	初期 (2011年1月10日日没)	31
4.2	日没時の計算プログラムを稼働させる前の時間の画像 (2011年1月20日 16:50 SH=1/800 , F=f/8 , ISO=200)。全体的に暗く、雲の視認が難しい。	33
4.3	改良後 (2011年2月6日日の出)	34
4.4	夜間 (暗夜) の雲の流れる様子 (2011年2月2日 SH=15 , F=f/3.2 , ISO=1600)。最初は全体にまだらに雲が広がっている様子が確認できる。時間を追って雲が流れていき、晴れていくのがわかる。また、全ての画像で雲の間に星が映っているのが確認できた。	35
4.5	左から雲のない日、薄雲の広がる日、厚い雲に覆われた日の画像。真ん中 (b) の画像は左 (a) と比べるとうっすらと雲が広がっているが、その下に星 (黄色四角) が映っているのがわかる。それに比べて右 (c) は全体に厚い雲で覆われている。	36
4.6	昼間の雲の流れる様子 (2011年2月3日 SH=1/800 , F=f/8 , ISO=200)。雲の形がはっきりと映しだされており、雲の流れもよくわかる。	37
4.7	左から雲のない日、薄雲の広がる日、厚い雲に覆われた日の画像。左 (a) のように快晴の日でもサチュレートすることはない。また、真ん中 (b) や右 (c) のように雲の厚さもわかる。	37
4.8	新月期 (2011年2月3日 5:24 月齢 0 SH=15 , F=f/3.2 , ISO=1600) の画像。	38
4.9	図 4.8 の黄色い部分の拡大図。ピンクの四角で囲んだのが確認できた限界等級で 5 等星。	38

4.10	新月期 (2011 年 2 月 4 日 5:24 月齢 28.7 SH=15 , F=f/3.2 , ISO=1600) の画像。画像の上の方には帯状の雲が確認できる。薄雲の下には星も確認できる。	39
4.11	半月 (2011 年 1 月 26 日 1:44 月齢 21.7 SH=15 , F=f/3.2 , ISO=1600) の画像。	40
4.12	図 4.11 の黄色い部分の拡大図。ピンクの四角で囲んだのが確認できた限界等級で 5 等星。	40
4.13	半月期 (2011 年 1 月 26 日 4:44 月齢 21.7 SH=15 , F=f/3.2 , ISO=1600) の画像。全体に雲がかかっているが、月の位置が確認できる。また、雲の薄い部分では、星も見える。	41
4.14	半月期 (2011 年 1 月 26 日 23:52 月齢 22.7 SH=15 , F=f/3.2 , ISO=1600) の画像。黄色の四角で囲んだ部分を拡大して限界等級を調べた。天気がよく、新月期より多く 5 等星を確認することができた。	41
4.15	満月期 (2011 年 1 月 20 日 2:26 月齢 15.7 SH=15 , F=f/3.2 , ISO=1600) の画像。	42
4.16	図 4.15 の黄色い部分の拡大図。月が明るすぎて、見える星が少ない。青の四角で囲んだ北斗七星がやっと確認できる。限界等級はピンクの四角で囲んだ 3 等星。	42
4.17	図 4.14 と同じ画像の色レベルを調整した画像。月により近い星を確認することができたが、限界等級は 3 等星。	42
4.18	満月期 (2011 年 1 月 20 日 3:12 月齢 15.7 SH=15 , F=f/3.2 , ISO=1600) の画像。全体にまだらに雲があるのが確認できる。露出時間の長さの分、雲の像が伸びて見えることもわかる。よって雲の形の判別は可能である。 . .	43
5.1	開発途中の観測環境情報を統合化した web サイト。	45

第1章 序論

1.1 かなた望遠鏡

かなた望遠鏡は広島大学宇宙科学センター附属東広島天文台に設置されている 1.5m 光学赤外線望遠鏡である。国内の望遠鏡の中でも最大級のもので、機動性にも優れ 1.5m の望遠鏡としては世界最速レベルの駆動速度を持っている。私達はこのかなた望遠鏡を用いて X 線衛星や γ 線衛星と密着した多波長観測研究を推進している。例えば γ 線バーストについては、衛星が天体を発見した際に発信される信号を受信後、なるべく早く (約 100 秒ほどで) 観測を開始することを目指している。かなた望遠鏡の性能、仕様を表 1.1 に示す。



図 1.1: かなた望遠鏡

光学系	リッチー・クレティアン光学系
主鏡の有効径	1540mm
主鏡材	ULE (超低膨張ガラス) 直径 1600mm 質量 983kg
合成 F 値と焦点距離 f	F=12.01 f=1850.7mm
視野	15 分角 (15 分角は月のみかけの半径程度)
焦点スケール	1mm あたり、投影した天球面上の 11.15 秒角
架台	経緯台 (トルク・モーターによるフリクションドライブ)
総重量	約 17 トン
搭載可能重量	500kg (カセグレン焦点)、1000kg (ナスミス焦点)
駆動速度	高度軸駆動:2 度/秒 方位軸駆動:5 度/秒

表 1.1: かなた望遠鏡の仕様

かなた望遠鏡による γ 線バーストの追跡観測の例に GRB 080506(図 1.2) がある。この観測から、可視・近赤外光は X 線とは別の機構で輝いている可能性が高いことが判明し、残光の物理メカニズムについて制限を与えた (Uehara, T. et-al, A&A, 519, A56(2010))。

このように、かなた望遠鏡を用いて人工衛星と連携した高エネルギー天体の多波長観測を推し進めている。

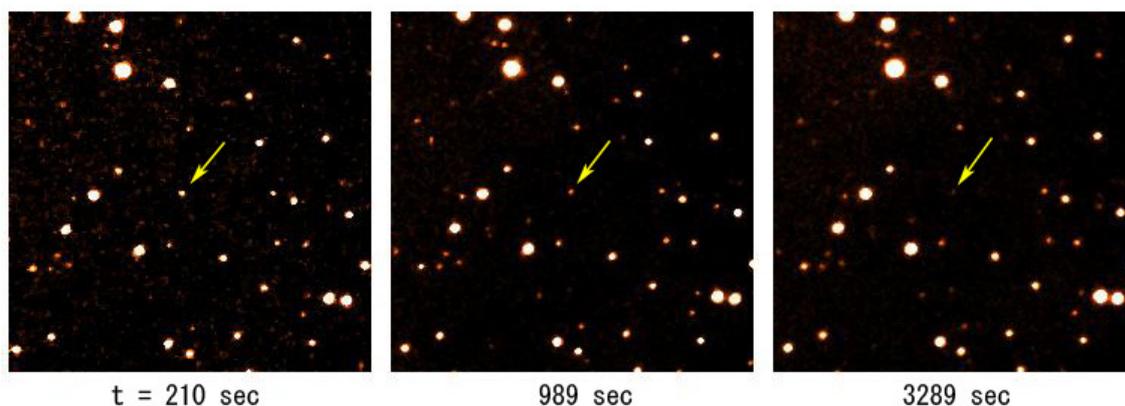


図 1.2: かなた望遠鏡と可視赤外線同時撮像分光装置 TRISPEC で観測した GRB 080506 の可視 V バンドの画像。左から順に γ 線で光出してから経過時間がそれぞれ 210 秒後、989 秒後、3289 秒後の画像であり黄色矢印で示した天体が次第に暗くなっていくのがわかる。

1.2 観測環境情報

観測環境情報とは、東広島天文台上空の気象状態を測定し、観測を行うのに適した状態かどうかを判断するための情報である。地上で観測を行うにあたって、空の透明度(雲)や、温度、湿度などの外的環境は観測精度や限界等級などに影響を与えることになる。そのため観測環境を把握し、さらに記録しておいて、後にデータのキャリブ等に役立てることができるようにしておくことは重要となる。観測者が屋外に出て観測環境を判断することも可能であるが、手間が掛かり、危険を伴う場合もあるという問題がある。また、人によって判断基準が異なり、正確な判断を行うには難しい場合もある。観測を効率的に行い、且つ観測データの質を定量的に評価するためには、望遠鏡の操作と同時に観測環境を定量的に測定することが重要である。

観測環境を調べる装置としては、湿度、風速、気温、気圧などを測定する気象センサーと雨を感知する雨滴センサー、雲の視認を行うスカイモニターがある。気象センサーと雨滴センサーから得られた情報を基に、現在の気象状態が望遠鏡のドームを開けて観測しても良い状態であるかどうかを主に判断し、スカイモニターで得られた画像は、主にどの天体が観測可能であるかを判断するものとなる。

さらに効率的な観測を行うためには、これらの観測環境情報とドーム内の様子や望遠鏡指方向の様子を撮影するビデオカメラなどの情報を一度に確認、判断することのできる

体制と、異常時にはわかりやすい警告を表示するようなシステムが必要とされる。

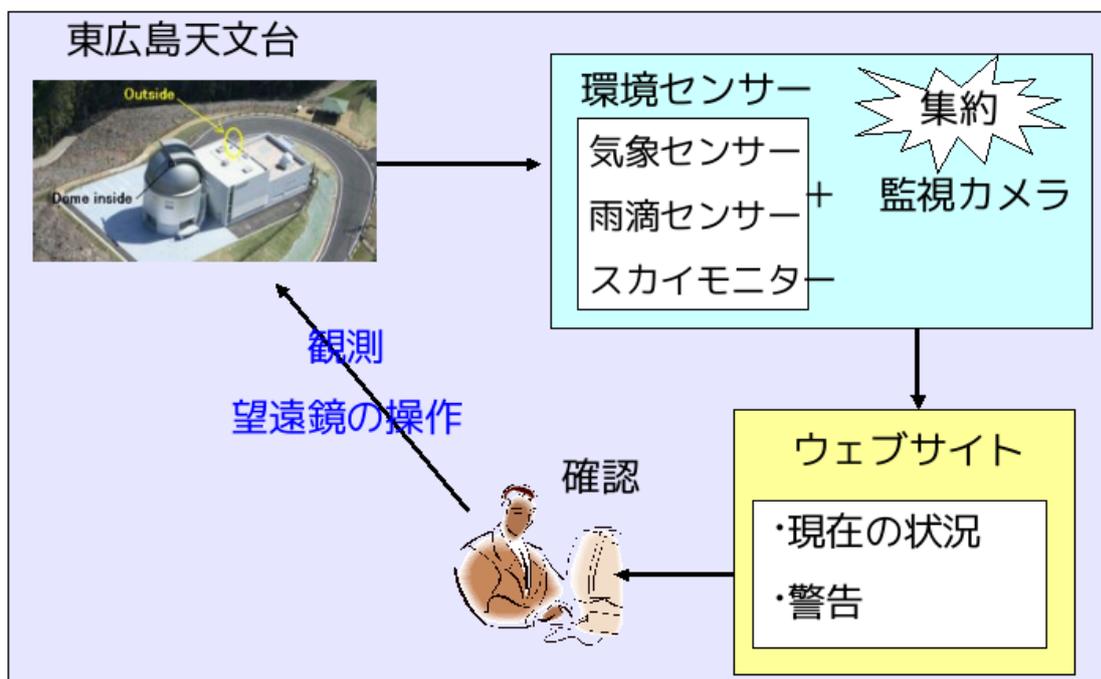


図 1.3: かなた望遠鏡による観測までの流れ。環境センサーは左上の東広島天文台の写真で黄色の丸で囲まれた部分に設置されており、そこで得られた情報を web サイト上に集約して公開し、それを確認して観測を行う。

図 1.3 はかなた望遠鏡で観測を行うまでの流れを模式的に表した図である。このように web サイト上で観測環境情報を確認することが出来れば、任意の場所から東広島天文台の観測環境を把握することができ、効率的な観測をできるようになるだけでなく、将来的に大学キャンパスの研究室からの遠隔操作による観測が可能になると期待できる。

1.3 スカイモニターとその要求仕様

本研究の主要なテーマとして、私はスカイモニターのアップグレードを行った。スカイモニターは、観測に特に大きな影響を及ぼす雲の視認をするためのものである。日常の観測では、スカイモニターで得られた画像を基に雲の位置を把握し、観測する天体を決定する。そのため、スカイモニターには天文台上空 (2π str) を全て収められるような広い視野と、リアルタイムに雲の位置や形を正確に視認するための短い時間インターバル、高い感度、さらに、暗夜から日の出へと時間が移り、空の明るさが変化しても、適切な露出パラ

メーターで空を映し出し、雲の視認性の度合も同じ状態を保つことができるような機能を兼ね備えたものであることが求められる。

このような要求を満たすことができれば、予測不能な突発的天体の検知、測光が可能となると期待される。過去に5等まで明るく光った γ 線バーストが確認されており、これは、スカイモニターによる検知も可能であると考えられる。

1.4 既存のスカイモニターと改善点

図 1.4 と図 1.5 は、既存のスカイモニターの検出部の図で、表 1.4 と表 1.4 はその仕様である(広島大学 2005 年度保田卒論参照)。これは、本研究のスカイモニターのアップグレードにおいて交換する部分である。



図 1.4: 既存スカイモニターの魚眼レンズ



図 1.5: 既存スカイモニターのカメラ (WAT-129NCCD カメラ)

メーカー	Fujinon
焦点距離	1.4(mm)
画角 (H × V)	185 ° × 185 ° (ϕ 4.6mm)
レンズしぼり	F/1.4 ~ F/16

表 1.2: 既存の魚眼レンズの仕様

メーカー	WATEC
有効画素数	768(H) × 494(V)
ユニットセルサイズ	8.4 μ m(H) × 9.8 μ m(V)
走査方向	2:1 インタレース
解像度	570TV 本
最低被写体照度	0.001Lux F1.4
AGC(ゲインコントロール) (S/N 比)	8 ~ 38 dB 52dB(OFF 時)
ガンマ特性	High \simeq 0.45 Low \simeq 0.6 OFF \simeq 1.0
積算時間	OFF,0.033,0.067,0.13,0.27, 0.53,1.1,2.1,4.3,8.5(sec) スイッチ切り替え式

表 1.3: 既存のカメラ (ビデオタイプ) の仕様

この魚眼レンズとモノクロカメラを用いた既存のスカイモニターが、かなた望遠鏡の移設とともに 2006 年から稼働している。このスカイモニターは安定して稼働し続けたものの、露出パラメータを自由に変更することができず、ダイナミックレンジが狭かった。それゆえに、ある程度の明るさがあるとサチュレートを起こし、昼間はもちろん、まだ観測可能な薄明時や満月の夜も、空の状態を見ることができない状態であった。

さらに、1 ピクセルあたりの画角が大きいため背景ノイズレベルが大きく、暗夜でも 2~3 等星くらいまでしか映すことができない。そのため、 γ 線バーストが起こっても検知、測光をすることは難しい。

この状況を改善するためには、どの時刻、天候においてもサチュレートを起こすことがないような、明るさに合わせた露出パラメータの設定で画像の取得を自動で行うプログラムの開発と、1 ピクセルあたりの画角が小さく、全天をカバーできるような分解能の高い (つまり有効ピクセル数が多い) カメラへの交換が求められる。

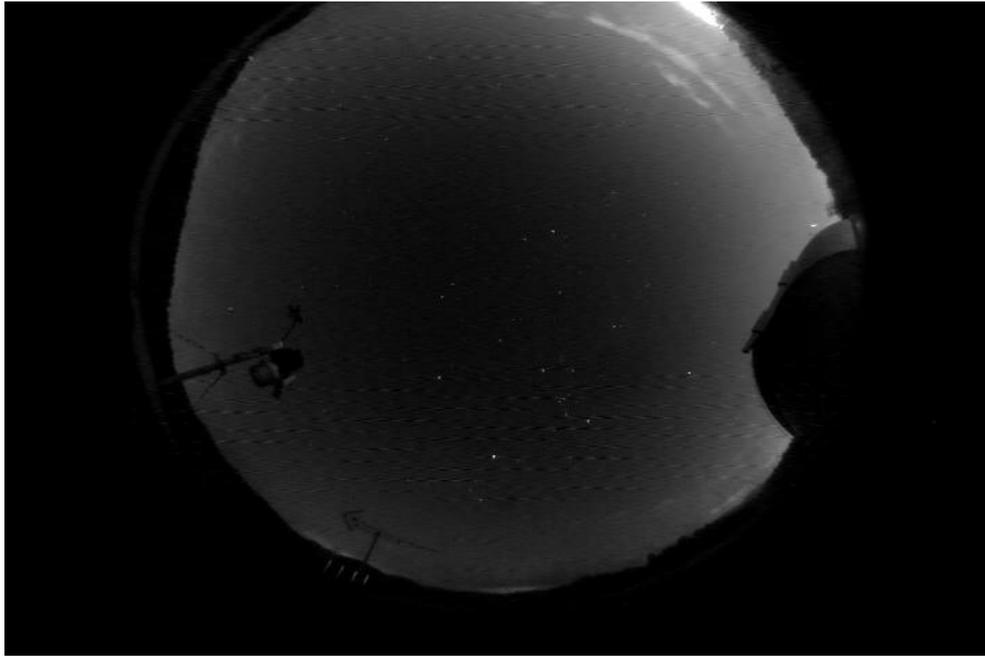


図 1.6: 既存のスカイモニターで撮影した画像 (2009 年 3 月 14 日)。画像処理をしない状態で確認できる限界等級は 3 等星で、色レベルの調整をした後では約 5 等星までが視認できた。

1.5 本研究の目的

今まで以上の効率的な観測へ向けた雲検知と、予測不能な突発的天体の検知を目標に、以下のように新しいスカイモニターの開発を進める。まず、既存のスカイモニターから、高感度で、広いダイナミックレンジを持ち、露出パラメーターを遠隔操作によって変更可能なカメラ (デジタルカメラ) への交換を行う。そして、既存のスカイモニターではサチュレートしてしまっていた時間帯でも、鮮明に空を映し出せるような露出パラメーターを決定し、時間 (太陽高度) に合わせた露出パラメーターを計算するプログラムを作成する。作成したプログラムで計算した、適した露出パラメーターで画像の取得を行い、取得した画像を加工、web 上での更新を自動で行うプログラムの開発を行う。このプログラムでは、かなた望遠鏡の観測データと共にスカイモニターで得たデータをアーカイブに保管するところまで担当させる。そして、どの時間帯でも雲が視認可能であるように露出パラメーターを調整しつつ、月の位相に応じた恒星像の限界等級の評価をする。

さらに、かなた望遠鏡による観測をより効率的なものとするため、現在別々に表示されている観測環境情報の統合化を行い、観測者が観測環境情報を一度に確認できるようにするとともに、集約した情報から観測可否の判定を自動的に行い、異常時には警告を表示させるようなシステムの開発を行う。

本研究によって、将来の東広島天文台の観測が、大学キャンパスにある研究室からの遠隔操作によって行われることを可能とする足がかりとしたいと考える。

第2章 ハードウェア

2.1 カメラ本体及び魚眼レンズ

スカイモニターのアップグレードにあたって、魚眼レンズとモノクロカメラを新しい魚眼レンズ(図 2.1)とデジタルカメラ(図 2.2)に交換した。



図 2.1: 新スカイモニターの魚眼レンズ (4.5mm F2.8 EX DC Circular Fisheye HSM) 図 2.2: 新スカイモニターのデジタルカメラ (NikonD3000)

大きく変更した点は、既存のものはビデオカメラ (WAT-120NCCD モノクロカメラ) であったのに対して、新しいものはデジタルカメラ (NikonD3000) に変更したという点である。デジタルカメラに変更することで、露出パラメーターの全てを PC からの遠隔操作によって変更可能となる。他の天文台においても同類のデジタルカメラが使用されており、問題なく稼働しているという実績も報告されている。

また、1pix 当たりのスカイの面積が約 1/10 になるので、恒星像に対するスカイレベルが 1/10 程度低減される。

ここで、恒星像は小さく、1pix にそのカウント全てが落ち込むとする。スカイ以外の

ノイズが無視できるものであるとすると、1pix 当たりのスカイレベルの等級は、

$$\begin{aligned}
 S/N &= 5 \\
 L_{limnew} &= 5N_{limnew} \\
 &= 5\sqrt{skyC_{new}} \\
 &= 5\sqrt{\frac{S_{new}}{S_{old}}skyC_{old}} \\
 &= \sqrt{\frac{S_{new}}{S_{old}}}L_{limold}
 \end{aligned}$$

となり、これを等級に直すと、

$$\begin{aligned}
 -2.5 \log_{10} L_{limnew} &= -2.5 \log_{10} \sqrt{\frac{S_{new}}{S_{old}}} - 2.5 \log_{10} L_{limold} \\
 M_{limnew} &= -2.5 \log_{10} \sqrt{\frac{S_{new}}{S_{old}}} + M_{limold} \\
 &= -2.5 \log_{10} \sqrt{\frac{0.087^2}{0.34 \times 0.38}} + 3 \\
 &\simeq 4.54
 \end{aligned}$$

L_{limnew}, L_{limold} :新、旧スカイモニターの 1pix 当たりの限界等級の光子数

$skyC_{new}, skyC_{old}$:新、旧スカイモニターの 1pix 当たりのスカイの光子数

S_{new}, S_{old} :新、旧スカイモニターの 1pix 当たりのスカイの面積

M_{limnew}, M_{old} :新、旧スカイモニターの限界等級

だけ暗くなるので、既存のスカイモニターの限界等級 3 等級に対して、新しいスカイモニターでは 4.5 等級程度まで限界が深まると期待される。

デジタルカメラ NikonD3000 は、国立天文台岡山天体物理観測所や和歌山県みさと天文台でスカイモニターに使用されている NikonD40 の後継機の中で、カメラの発注時に最も安価なものであった。また、魚眼レンズ 4.5mm F2.8 EX DC Circular Fisheye はデジタルカメラ NikonD3000 に適用するもので、全天 ($2\pi\text{str}$) の撮影が可能であるためスカイモニターに使用するのに適したものである。デジタルカメラを使用するにあたって、シャッター機構の寿命が問題となるが、スカイモニターにデジタルカメラを仕様している他の天文台によると、定点で同間隔で撮影していることから、シャッター機構の寿命を越えて、3 年以上の使用が可能であることがわかっている。

このデジタルカメラは、有効画素数が 3872×2592 画素である。既存のモノクロカメラは 768×494 画素であったので、カメラを交換することで星像イメージに対するスカイノイズが小さくなることが期待できる。また、USB ケーブルによって PC との接続が可能で、フリーウェア gPhoto2 を使用することで PC でのカメラの制御が可能となる。これに

よって、遠隔操作や、自動プログラムによる露出パラメーターの設定変更、画像の取得が可能となり、どの時間においても最適なパラメーターでの撮影が可能となる。よって、今までサチュレートを起こしてしまっていた時間帯、すなわち満月や、薄明時、さらに昼間であっても雲の検知が可能となると考えられる。新しいスカイモニターに仕様する魚眼レンズ、デジタルカメラの仕様は次の表 2.1 と表 2.2 に示す。

メーカー	Nikon
焦点距離	4.5(mm)
画角	180 °
レンズしぼり	F/2.8 ~ F/22

表 2.1: 魚眼レンズの仕様

メーカー	Nikon
有効画素数	10.2 メガピクセル
撮像素子	23.6 × 15.8mm サイズ CCD センサー
記録画素数	3872 × 2592 ピクセル
シャッター型式	電子制御上下走行式フォーカルプレーンシャッター
シャッタースピード	1/4000 ~ 30.0 秒 (1/3 ステップ)
ISO 感度	100,200,400,800,1600,3200
インターフェース USB	Hi-Speed USB
カメラの大きさ (W × H × D)	約 126 × 97 × 64 mm
カメラの質量	約 485g (バッテリー本体、SD メモリーカード、ボディキャップを除く)
動作環境 温度	0 ~ 40
動作環境 湿度	85 %以下

表 2.2: デジタルカメラ NikonD3000 の仕様

新しいスカイモニターで用いる治具、つまりデジタルカメラや、魚眼レンズを固定する治具は前任者 (広島大学大学院理学研究科物理科学専攻 奥嶋貴子氏) が設計し、理学部金工室に依頼して製作したものである。

2.2 ハウジングと周辺機器

スカイモニターは東広島天文台の屋上に設置してある。このため、湿気、結露、降雨などに対する対策と温度調整が必要である。既存のスカイモニターは、このような気象条件

に対応できるよう、気象用密閉ケースの中に温度計やヒーターと共に収納されている。このスカイモニターは、設置から現在まで5年間メンテナンスなしで極めて安定して稼働し続けている。よって、カメラ以外の筐体関係、カメラを収納する気象用密封ケースやヒーター等は、引き継いで使うこととした。引き継いで使う物品の写真とその仕様を以下に示す(広島大学 2005 年度保田卒論参照)。



図 2.3: 気象用密封ケース

材質	プラスチック
寸法 (W × H × L)	290×190×340(mm)

表 2.3: 気象用密封ケースの仕様

材質	アクリル
メーカー	はざい屋
形状	厚み 2mm 外径 100φ フランジ巾 10mm

表 2.4: 透明アクリルドームの仕様



図 2.4: OMRON 温度制御用デジタルサーモ



図 2.5: 坂口電熱シリコンラバーヒーター



図 2.6: DC ファンモータ

図 2.3 に示すのが、装置全般を収納する密封ケースである。気象用に作られた密封性の高いケースで、監視用の半球の透明アクリル製ドームを接着したものである。ヒーターにはシリコンラバーヒーター(図 2.5)を使用し、結露対策としている。さらに効率的な暖房をするためにファン(図 2.6)を取り付けており、デジタル温度コントローラ(図 2.4)でスイッチの ON/OFF 切替えを可能としている。ファンは 12V 電源(図 2.7)に接続されている。この 12V 電源の入力電圧は 100V で、表 2.5 はその仕様である。ヒーターと温度コントローラは AC100V 入力である。



メーカー	DENSEI-LAMBDA 社
定格直流出力電圧	12V
定格直流出力電流	4.4A
入力電流	1.2A (100VAC)
動作周囲温度	-10 ~ +60
冷却方法	自然空冷
質量	450g

図 2.7: 12V 電源 EWS50-12

表 2.5: 電源 EXS50-12 の仕様

図 2.8 に示すのが、既存のスカイモニターの装置図である。破線で囲まれた部分が気象用密封ケースの内部である。赤い丸で囲んだ部分が既存のスカイモニターの検出部で、本研究において新しいデジタルカメラと魚眼レンズに交換する部分である。

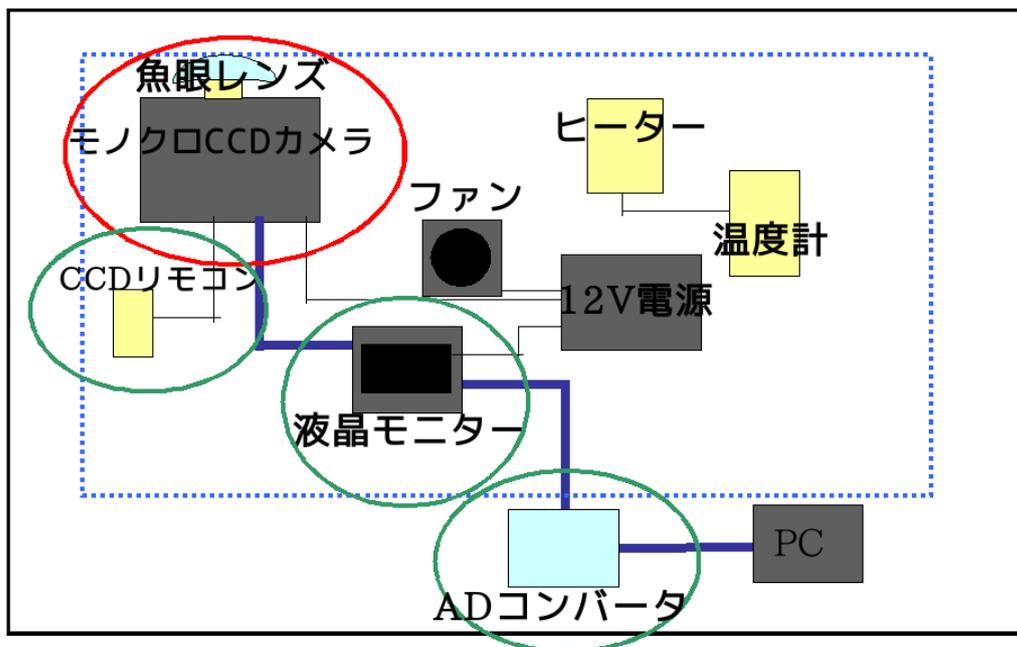


図 2.8: 既存スカイモニターの装置図

図中で緑の丸で囲んだ CCD リモコンは、モノクロカメラを制御するためのリモコンであり、液晶モニター、AD コンバーターはモノクロカメラで得られた映像を PC で読み取れるように変換するための機器である。これらは、モノクロカメラを用いるために必要な

機器であり、今回デジタルカメラに交換することで必要なくなるため、デジタルカメラの設置に伴い取り外した。

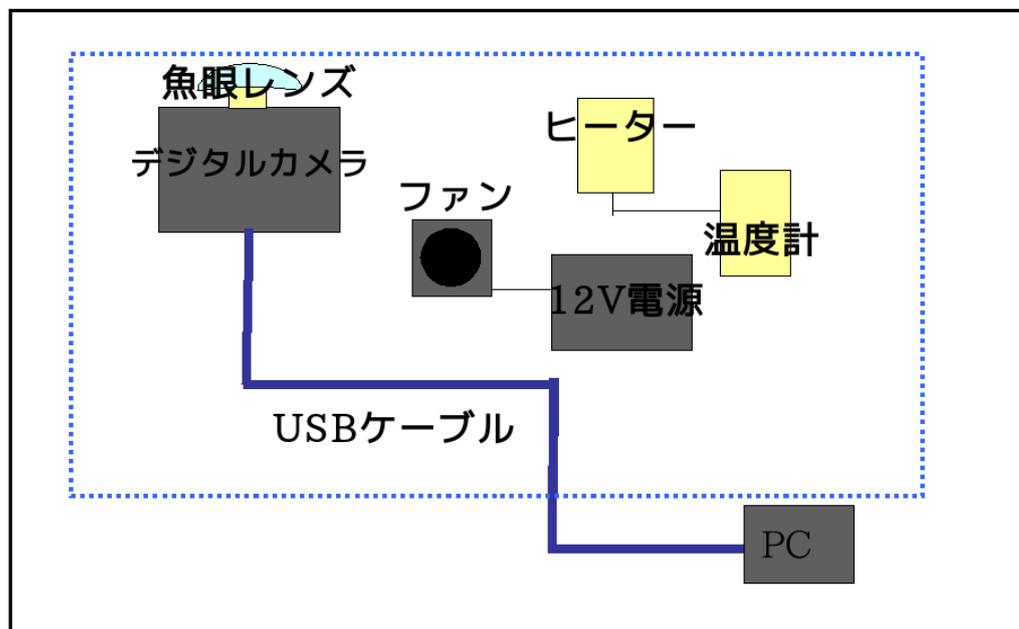


図 2.9: 新スカイモニターの装置図

図 2.9 は、モノクロカメラを新しいデジタルカメラに交換し、CCD リモコン、液晶モニター、AD コンバーターを取り外した後の装置図である。新しいスカイモニターの検出部のデジタルカメラは、USB ケーブルと USB リピータケーブル (5m×5 本) を通して PC とつながっており、カメラ交換後のスカイモニターは PC による制御、遠隔操作が可能となる。

2.3 カメラの設置及び PC との接続

既存のスカイモニターに使用されている、モノクロカメラや周辺機器は、密封ケースの中に収納され、中のアルミ基盤に装着されている。図 2.10 に示すのが、既存のスカイモニターのアルミ基盤である。図 2.10 がそのアルミ基盤であり、緑色の台がモノクロカメラを固定している光学台である。

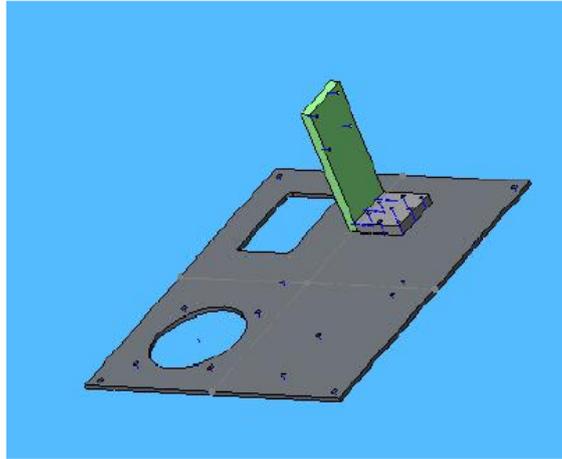


図 2.10: 既存スカイモニターのアلمミ基盤と光学台の合成図

新しいデジタルカメラの設置にあたって、レンズホルダーの外径が大きくなり、アクリルドーム部の開口に余裕がなくなって干渉する可能性があるため、デジタルカメラのレンズの中心とアクリルドームの中心を1~2 mmの誤差で合わせる必要があった。その要求を満たすように、アアルミ基盤や光学台を加工した。施した加工は、光学台に「レ」の字型のカメラを支えるための台を取付け、光学台を既存の位置より後(四角い穴と遠ざかる方向)にずらすということと、その台を後から微調整が可能ないようにネジ穴をつけ直すというものである。

カメラを支えるための台は、前任者(奥嶋貴子氏)が適切な大きさ、形(「レ」の字型)に設計し、準備してくれていたもので、私はこの台を固定・調整するため、既存のアアルミ基盤及び光学台に加工を施した。

カメラを取り付けるための台は光学台に取付け、全天を映し出すのに適切な高さに設置する。光学台にこの台を固定するためのネジ穴と、デジタルカメラが適切な高さに来る位置に光学台をアアルミ基盤に固定するネジ穴を空け、それぞれをネジで固定した。デジタルカメラをネジで固定するだけでは、デジタルカメラがしっかりと固定されなかったため、カメラを取り付ける台をビニルテープで巻くことでデジタルカメラを動かないように固定した。

また、新たにデジタルカメラとPCをUSBケーブルでつなぎ、PCで制御することになったことを受け、スカイモニターを設置する制御棟屋上から制御用PCのある2階の制御室まで、USBリピータケーブル(5m×5本)をはわせた。USBリピータケーブルの使用については、20m(5m×4本)まで可能であることは確認されていたが、20mでは、制御室までとどかないため、カメラ設置前に動作確認を行い、問題なく動くことを確認してから作業を行った。また、スカイモニターは屋外に設置されているため、USBケーブルの一部は屋外で風雨にさらされることになる。この対策として、USBケーブルの紫外線対策(テープを巻く、ないしはモールへ入れる)及び、雨漏り対策(自己融着テープを巻

く、ないしコーティング剤で密封する)を施した。

図 2.11 に示すのは、モノクロカメラをデジタルカメラに交換した後の筐体内部の写真である。



図 2.11: デジタルカメラに交換した後の筐体内部

第3章 ソフトウェア

3.1 ソフトウェアへの要求仕様

スカイモニターで得られる画像は、雲の状態を判断し、どの天体を観測するかを決めるための材料となる。効率的な観測を行うためには、望遠鏡の操作と同時にリアルタイムの雲の情報をスカイモニターで得る必要がある。そのため、短い時間間隔で自動的に画像を取得し、安定的に更新をし続けることのできるソフトウェアであることが求められる。

さらに、既存のスカイモニターではサチュレートして観測不可能であった、薄明時や満月、昼間であっても雲検知を可能にするため、時刻(太陽高度など)に合わせた露出パラメーターの計算、設定変更を遠隔から自動で行うソフトウェアの開発が必要とされる。

また、得られた画像を見易くするために、画像を回転・トリミングさせたり、どのようなパラメーターで得た画像かがわかるように文字の入力などを行うなど、撮ったままの画像ではなく、見易い画像に加工することもソフトウェア中の機能に含むことが必要である。

3.2 C言語プログラムによる自動画像取得プログラムの作成

ソフトウェアの開発は、Linuxマシン上でC言語を用いて行った。作成したプログラムでは、その時間に適したパラメーターを算出し設定後、画像の取得を行い、得られた画像を加工し、完成した画像をWeb上に更新、アーカイブに保存までを自動で行うことができるようにした。

私の作成したプログラムでは、まずパソコン内時計の時刻の取得を行い、その時間において最適な露出パラメーターを計算(計算プログラムについては後で述べる)、計算した露出パラメーターで画像を取得し加工するという流れで処理が進むようになっている。このうち加工の内容は次のとおりである。



図 3.1: スカイモニター (手前) と西にあるかなた望遠鏡のドーム (奥)



図 3.2: 加工前の画像。2011 年 1 月 27 日 16:30 に撮影した画像の生データで、東西南北が一致しておらず、左上にかなた望遠鏡のドームとその横に夕日が見える。また、左右の黒い部分は不要。

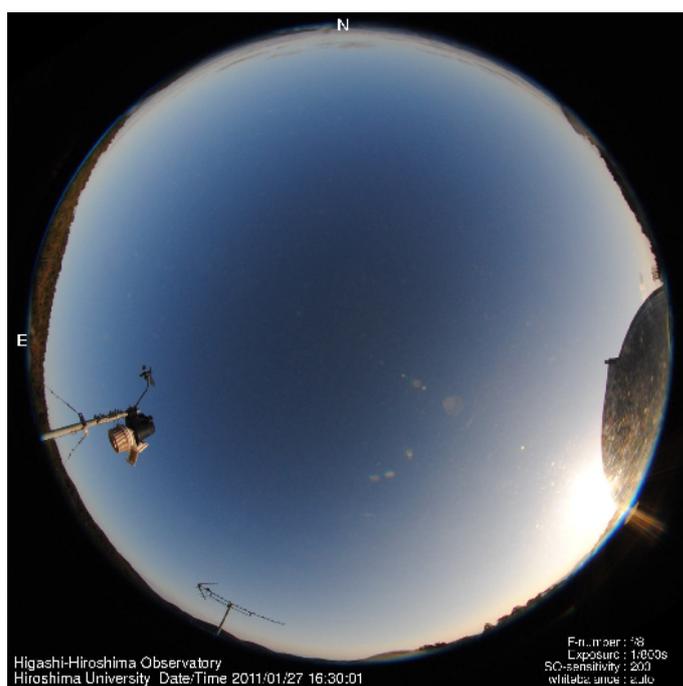


図 3.3: 加工後の画像 (2011 年 1 月 27 日 16:30)。方角が正しく調整され、余分な部分が切り取られている。左下には撮影時刻、右下には露出パラメータを書き込んだ。

図 3.1 は屋上に設置したスカイモニターを西向きに映した写真でスカイモニターの西側にかなた望遠鏡のドームがあることがわかる。図 3.2・図 3.3 は 1 月 27 日 16:30 に撮った画像で、加工前の生データと加工後の画像である。方角も合っておらず、余分な部分の多い図 3.2 と比べ、図 3.3 は方角も正しく、余分な部分は切り取られている。これは、かなた望遠鏡の望遠鏡制御ソフト上の全天図とよく対応していて、見比べるのが容易である。さらに撮影時刻や、露出パラメータも示して、わかりやすい。この画像の最終的な画像サイズは 2200pix×2200pix(JPEG フォーマット) である。

このように、加工された画像は、日付や時間のわかるファイル名を付けて、日ごとに作成されるディレクトリ内に保存される。また、Web 用には、Web に掲載するのに適したサイズ (600pix×600pix) へ画像を圧縮し、Web 上のデータの更新を自動で行うプログラムを作成した。

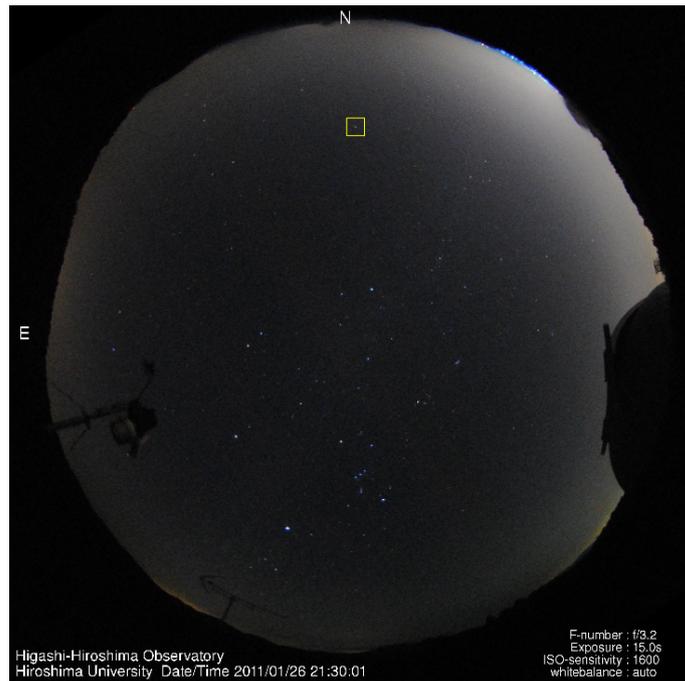


図 3.4: 加工後の夜の画像 (2011 年 1 月 26 日 21:30)。「N」の下の黄色い丸は北極星で、下の方にはオリオン座も確認できる。

図 3.4 は作成したプログラムで取得した夜の画像の例である。この図からわかるように、方角が調整されている。また、左下に撮影時刻、右下には露出パラメーターがプリントされ、加工前の画像 (図 3.1) と比べて実用性の高い画像に加工できている。作成したプログラムのソースは巻末の付録に掲載する。

デジタルカメラを PC で制御するにあたって、フリーウェアの gPhoto2(2.4.9.1) を使用し、取得した画像の加工には、ImageMagick を使用した。

デジタルカメラの制御 : gphoto2(2.4.9.1)

gPhoto2 は Linux や Unix 系のシステムにおいてデジタル写真を処理することのできるフリーのソフトウェアである。画像をカメラから転送したり、PC からのリモートコントロール、キャプチャーなどの機能を使うことができる。また、カメラの制御をするために用いたコマンドは C 言語のプログラムに組み込み、プログラムからカメラの制御を行うこともできる。

カメラの制御に使用するコマンドの例

- gphoto2 -get-config=STRING (設定値の取得)

- `gphoto2 -set-config=STRING` (設定値の設定)
- `gphoto2 -capture-image-and-download` (画像の取り込みとダウンロード)

画像の加工 : ImageMagick

ImageMagick は画像を操作したり、表示するためのソフトウェアである。これは、デジタルカメラで画像を取得したときに得られる画像ファイル、JPG ファイルにも対応している。また、コマンドラインから利用できるだけでなく、C 言語を用いたプログラムからも利用できるの、本研究において画像の加工を行うのに、この ImageMagick を使用した。

画像の加工に使用するコマンドの例

- `convert -rotate +90 magick.jpg magick-rotate.jpg`
(画像を時計回りに回転)
- `convert -crop WxH+X+Y magick.jpg magick-crop.jpg`
(画像を (X,Y) から $W \times H$ の大きさに切取る)
- `convert -font helvetica -fill COLOR -pointsize SIZE -draw'text X,Y "文字"' magick.jpg magick-font.jpg`
(画像の (X,Y) の位置に”文字”を入力する)
- `convert -geometry WxH magick.jpg magick-geometry.jpg`
(画像を $W \times H$ の大きさに変更)

3.3 最適露出パラメーターの計算法

露出パラメーターはその時点での太陽高度に合わせて最適な設定に変更する必要がある。今回作成するプログラムで変更する露出パラメーターは、露出時間、F 値、ISO 感度の 3 種類である。

露出時間はシャッターを開放する時間で、撮像素子がレンズを通した光にさらされる時間である。F 値は絞りの大きさを示す。正確には、レンズの焦点距離を有効口径で割った値で、像の単位面積あたりの光量は F 値の 2 乗に反比例する。ISO 感度は写真フィルムの規格であり、あるフィルムがどの程度弱い光まで記録できるかを示す。一段高い ISO 感度を使用することで、露出時間を一段速くするか絞りを一段絞ったのと同じ効果が得られるが、背景ノイズが強調される欠点を抱えており、最終的には見易さを人間が判断して決めることとした。

最適露出パラメーターの導出については、日の出、日没の前後以外の時間は、昼間は昼間の設定で固定、夜間は夜間の設定で固定し、日の出、日没前後は日の出、日没時との時

間差に応じた設定となるような計算式(後述)を考え、試験観測の上、微調整を行い実際に値するものとした。この計算プログラムもC言語を用いて作成した。

私の作成した計算プログラムでは、まず現在の日時を取得し、その日の日の出、日没時刻を計算する。この計算は天文分野で広く用いられているC言語ライブラリを持ち込んだ。そして、現在時と日の出、及び現在時と日没との差を求め、その二つの差から設定するパラメーターを計算する。開発段階では、天文薄明(太陽高度-18°)開始、終了の目安となる日の出の90分後から日没の90分前までは昼の設定、日没後90分から日の出の90分前までは夜の設定とし、日の出、日没の前後90分の間はある式に基づいて設定を決定することにした。この数値は後の試験観測で調整することになる。

計算式の関数形であるが、日の出、日没時が空の光量が一番変化する時間であると考えられるため、原点(日の出、日没時)の傾きが一番急で原点から離れるにつれて、傾きが0になり、昼間ないし夜の固定値になだらかに繋がるような関数が良いだろうと考え、sin関数を採用した。

F値を例に上げると、昼のF値の固定値を F_{day} 、夜の固定値を F_{night} とし、現在時を t (分)、日の出時刻を T_{rise} (分)、日没時刻を T_{set} (分)としたとき、日の出前後のF値(F)を求める式は、

$$F = -\frac{F_{night} - F_{day}}{2} \sin\left(\frac{2\pi}{2 \times (90 + 90)} \times (t - T_{rise})\right) + \frac{F_{night} + F_{day}}{2} \quad (3.1)$$

と仮定した。日没時はこの逆となるよう、プラスとマイナスを逆にした。それが次の式である。

$$F = \frac{F_{night} - F_{day}}{2} \sin\left(\frac{2\pi}{2 \times (90 + 90)} \times (t - T_{set})\right) + \frac{F_{night} + F_{day}}{2} \quad (3.2)$$

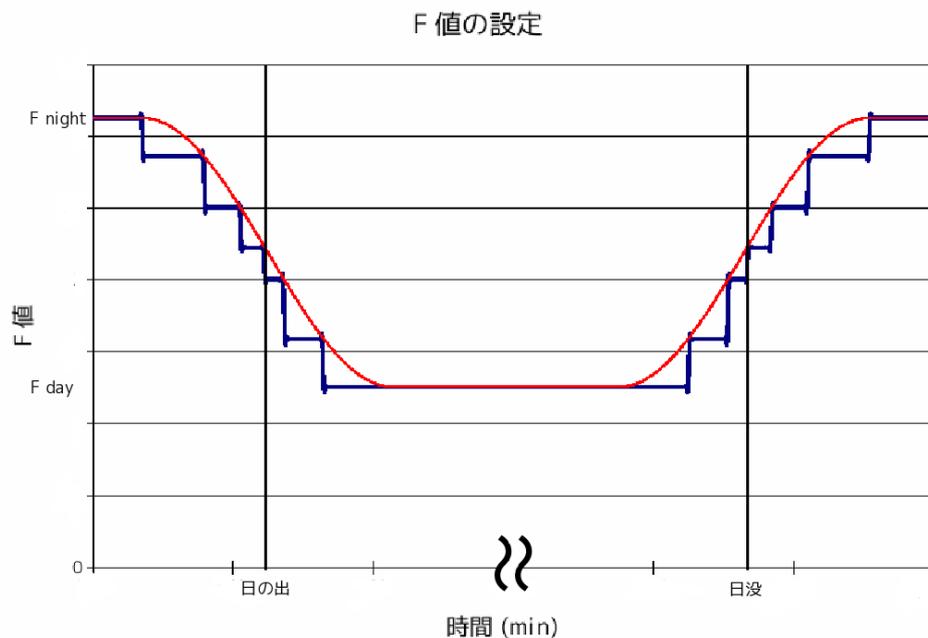


図 3.5: F 値設定のモデル曲線 (初期) のグラフ。縦軸は F 値、横軸は時間 (分) を表す。横軸の目盛は 150 分間である。赤線は計算による理想の値、青線は実際に設定する値。

図 3.5 はこの仮定を基に作成したプログラムで、時間毎に設定する F 値を示すグラフである。赤線で示した関数値はなだらかであるが、カメラで設定できる値はとびとびの値なので、実際には青線で示した値に設定するプログラムにした。その際、サチュレートを起こすことがないように、大きめの値を取るよう設定することにした。露出時間、ISO 感度についても同様に設定した。ISO 感度 (3.6)、露出時間 (3.7) のグラフを次に示す。

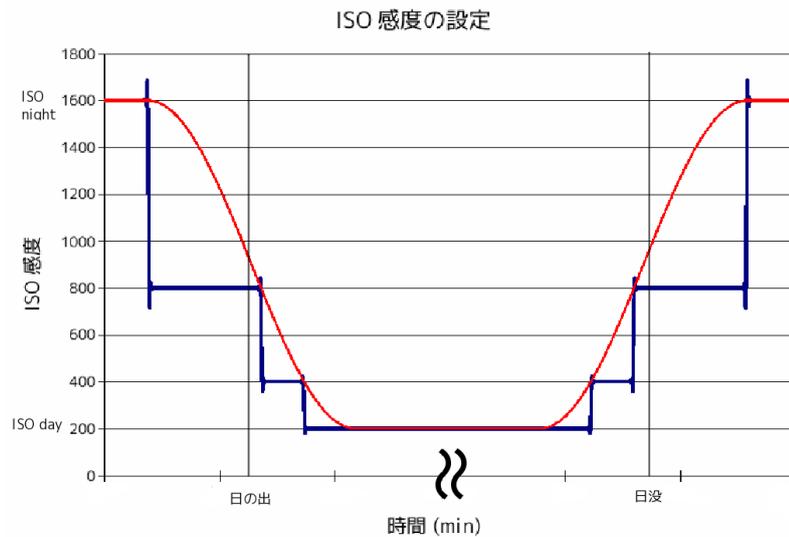


図 3.6: ISO 感度設定のモデル曲線 (初期) のグラフ。縦軸は ISO 感度、横軸は時間 (分)、1 目盛は 150 分を示す。赤線は計算による理想の値、青線は実際に設定する値。

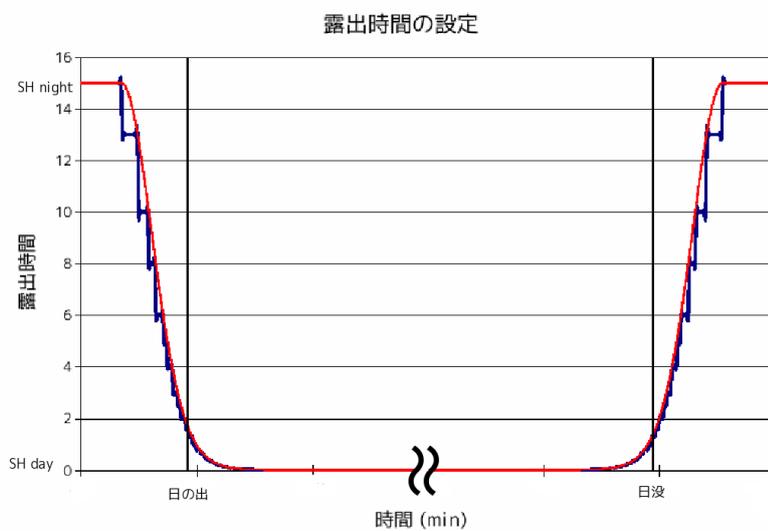


図 3.7: 露出時間設定のモデル曲線 (初期) のグラフ。縦軸は露出時間、横軸は時間 (分)、1 目盛は 150 分を示す。赤線は計算による理想の値、青線は実際に設定する値。縦軸には露出時間の対数 (log) を使用した。

露出時間の設定のグラフ (3.7) のグラフは sin 関数の形からは崩れているのがわかる。つ

まり、日の出、日没時が設定の中心となるようになっていない。これは、露出パラメータの中で最も重要である露出時間については、他のパラメータのように、縦軸にリニアの値を使用せず、露出時間の対数 (log) を使用しているからである。つまり、露出時間の対数値が sin 関数に従って変化するとして露出時間を求めている。当初、リニアな値を使用したが、その場合日の出や日没の前後で、明らかに露出オーバーとなったり、露出不足となったりするケースが目立った。これに対応させるため、露出時間の対数をとることにした。図 3.8 はリニアのままの関数値のグラフ (青線) と対数をとった関数値のグラフ (赤線) を比較している。日の出、日没前後で違いが大きいことがわかる。

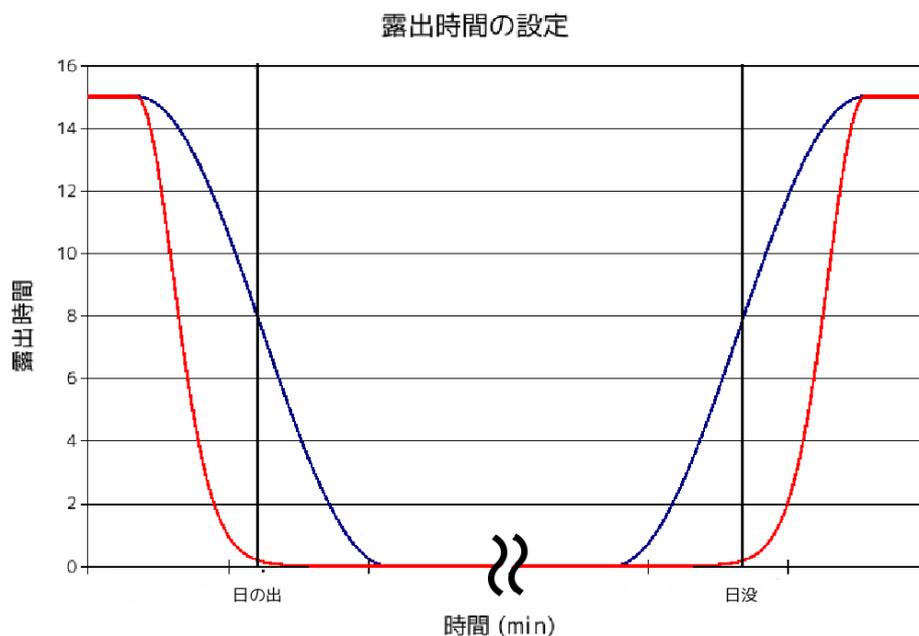


図 3.8: 露出時間のリニアと対数のグラフ。青線はリニアのままのグラフで、赤線は対数をとったグラフである。日の出、日没前後で違いが大きい。

それぞれの露出パラメータでこの関数を基に計算プログラムを作成し、試験観測を行い、実用化に向けて調整をした。

最終的に採用した昼間と夜間の露出パラメータは、それぞれ次の通りである。

日の出前後の露出時間 SH は、

$$SH = - \frac{\log_{10}(SH_{night}) - \log_{10}(SH_{day})}{2} \sin \left\{ \frac{2\pi}{2 \times (50 + 25)} \times \left((t - T_{rise}) + \frac{50 - 25}{2} \right) \right\} + \frac{\log_{10}(SH_{night}) + \log_{10}(SH_{day})}{2} \quad (3.3)$$

で、日没の前後の露出時間は、

$$SH = \frac{\log_{10}(SH_{night}) - \log_{10}(SH_{day})}{2} \sin \left\{ \frac{2\pi}{2 \times (50 + 25)} \times \left((t - T_{set}) + \frac{25 - 50}{2} \right) \right\} + \frac{\log_{10}(SH_{night}) + \log_{10}(SH_{day})}{2} \quad (3.4)$$

である。

F 値の日の出 (式 (3.5))、日没 (式 (3.6)) 前後は、

$$F = -\frac{F_{night} - F_{day}}{2} \sin \left\{ \frac{2\pi}{2 \times (50 + 60)} \times \left((t - T_{rise}) + \frac{50 - 60}{2} \right) \right\} + \frac{F_{night} + F_{day}}{2} \quad (3.5)$$

$$F = \frac{F_{night} - F_{day}}{2} \sin \left\{ \frac{2\pi}{2 \times (50 + 60)} \times \left((t - T_{set}) + \frac{60 - 50}{2} \right) \right\} + \frac{F_{night} + F_{day}}{2} \quad (3.6)$$

ISO 感度の日の出 I(式 (3.7))、日没 I(式 (3.8)) 前後は、

$$I = -\frac{I_{night} - I_{day}}{2} \sin \left\{ \frac{2\pi}{2 \times (25 + 50)} \times \left((t - T_{rise}) + \frac{50 - 25}{2} \right) \right\} + \frac{I_{night} + I_{day}}{2} \quad (3.7)$$

$$I = \frac{I_{night} - I_{day}}{2} \sin \left\{ \frac{2\pi}{2 \times (50 + 25)} \times \left((t - T_{set}) + \frac{25 - 50}{2} \right) \right\} + \frac{I_{night} + I_{day}}{2} \quad (3.8)$$

である。

露出パラメーターを変える時間帯については、当初日の出、日没の前後 90 分間としていたが、実際に運用を続けてみたところ、見にくくなる時間帯があった。このことから、変更する時間帯を暗い時間帯にシフトさせる必要があると考え、最終的には露出時間と ISO 感度は日の出の 50 分から 25 分後、日没の 25 分前から 50 分後、F 値は日の出の 50 分前から 60 分後、日没の 60 分前から 50 分後を考えた計算式に基いて変更し、それ以外の時間は昼間ないし夜の固定値とした。F 値だけ変更する時間帯を変えたのは、全て同じにすると日没 30 分前あたりの時間の画像が少し暗くなっていた。露出時間と ISO 感度の時間帯を変えると他の時間が見にくくなるため、F 値を早い時間から変更することで解決した。

第4章 性能評価と考察

4.1 試験観測

スカイモニターを新しいカメラに交換し設置した後、作成した自動画像取得プログラムを一定の間隔で稼働させ試験観測を行った。

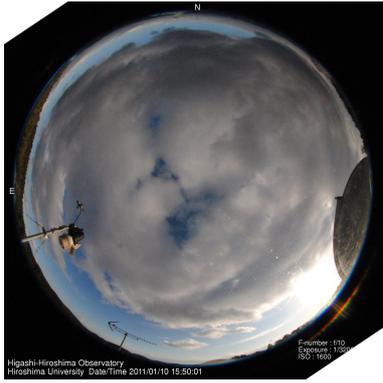
昼や夜の様々な時間の画像を取得し、各時間における雲の視認度合を確認し、必要に応じて調整した。また、日の出、日没の露出パラメーターの計算プログラムが実用に値するものとなるように調整を行った。さらに様々な月のフェーズをカバーする期間試験観測を行い、月の位相に応じた恒星像の限界等級の評価を行った。

リアルタイムに雲の状態を監視したいので、出来るかぎり短い時間間隔での撮影を目指した。画像を取得するコマンドを打ってから、画像の取得を終えるまでに30秒以上の時間が必要であることはわかっていたので、プログラムは1分間隔で動かすこととした。しかし、1分では、先に起動したプログラムが終わる前に次のプログラムを起動してしまうことがあり、同じ画像を更新していたり、時には、カメラがハングアップを起こしてしまうことが数回あった。そのため、プログラムは2分毎に起動するものとした。

4.2 薄明時の露出パラメーター

第3章で述べたように、開発した計算プログラムを用いて薄明時(日の出、日没時)の観測を行った。

初期の日の出、日没前後90分間設定を変化させるプログラムで撮影した日没時の画像が、次の図4.1の12個の画像である。このときは、まだISO感度の設定は変更させておらず、また昼の設定が露出時間は $1/3200\text{s}$ 、F値は $f/10$ で、夜は露出時間 15s 、F値 $f/3.2$ に設定していた。図では露出時間をSH、F値をF、ISO感度をISOとした。



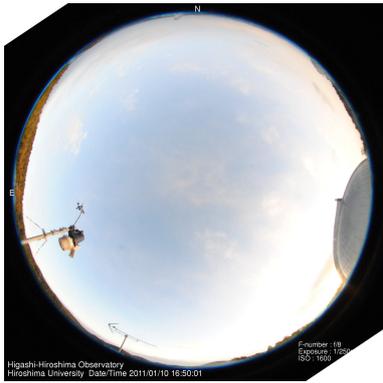
(a) 15:50 (SH=1/3200 , F=f/10 , ISO=1600)



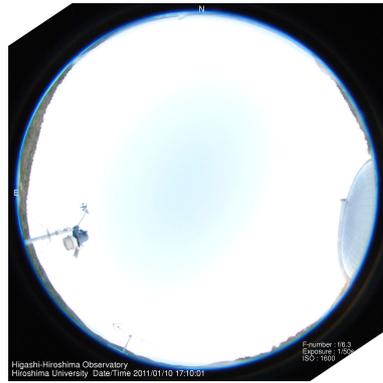
(b) 16:10 (SH=1/2500 , F=f/10 , ISO=1600)



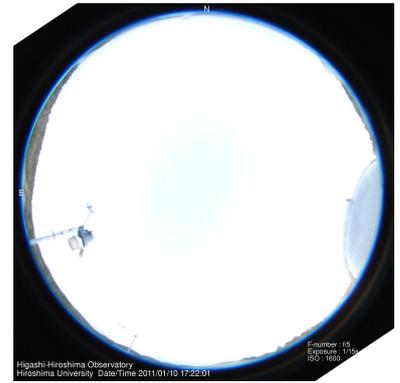
(c) 16:30 (SH=1/1600 , F=f/9 , ISO=1600)



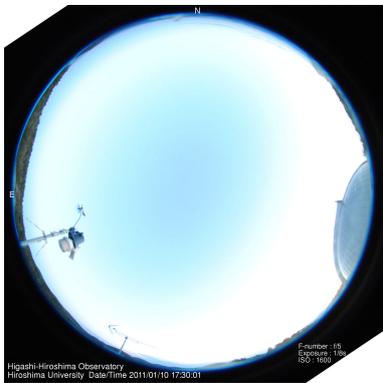
(d) 16:50 (SH=1/2500 , F=f/8 , ISO=1600)



(e) 17:10 (SH=1/50 , F=f/6.3 , ISO=1600)



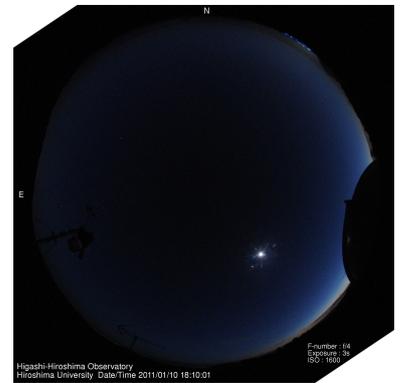
(f) 日没 17:22 (SH=1/15 , F=f/15 , ISO=1600)



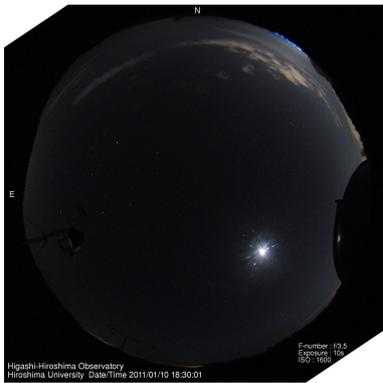
(g) 17:30 (SH=1/8 , F=f/5 , ISO=1600)



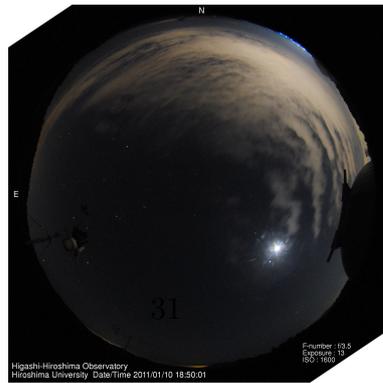
(h) 17:50 (SH=1/1.3 , F=f/4 , ISO=1600)



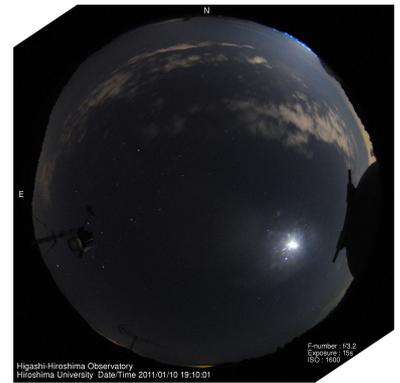
(i) 18:10 (SH=3 , F=f/4 , ISO=1600)



(j) 18:30 (SH=10 , F=f/3.5 , ISO=1600)



(k) 18:50 (SH=13 , F=f/3.5 , ISO=1600)



(l) 19:10 (SH=15 , F=f/3.2 , ISO=1600)

図 4.1: 初期 (2011 年 1 月 10 日日没)

図 4.1 から、(a,b,c) では、雲の形など視認が可能であるが、明るくなりすぎている時間帯 (d,e,f,g) や、暗くて見辛い時間帯 (j,k) があり、この時間帯での雲の視認は十分にできない。また (c) はもう少し明るい時間に対応する設定 (露出時間を短くするか絞りをしぼる) の方が鮮明になると思われ、(i,j,k) では、暗い時間に対応する設定 (露出時間を長くする、または絞りを開く) にすると見える星の数が増えると考えられる。

このように、当初仮定した、日の出、日没の前後 90 分間設定を変更させるプログラムのままでは、実用するには不十分なものであった。これを解決させ、実用に値するものにするためには、ISO 感度も露出時間や F 値と同様に設定を変更させ、計算プログラムに基づいて設定を変更させる時間を短くし、さらに、sin 関数の中心 (最初のプログラムでは日の出、日没時) をそれぞれ暗い方の時間帯にシフトさせる必要があると考えた。

そこで、3つのパラメーターを変更させる時間帯を、暗い時間帯にシフトさせて結果をみることにした。日没の 15 分前から 45 分後、20 分前から 50 分後など様々な時間帯で設定変更させてみた結果、最終的に日没の 25 分前から 50 分後、日の出では 50 分前から 25 分後を計算プログラムによって設定変更することで、その時間帯は、図 4.1(d,e,f) のように明るくなりすぎて雲の視認ができない、ということが起こらないようにすることができた。

ただし、図 4.2 のように、日没時の計算プログラムを稼働させる前の数分と、日の出の計算プログラムを稼働させた後の数分が、実際の空と比べて暗くなって雲の視認が少し難しくなってしまった。これを解決するために、F 値の設定だけ、日没時は早めに変更を始め、日の出時は遅くに昼の設定になるように、日没の 60 分前から 50 分後、日の出の 50 分前から 60 分後に設定を変更するよう時間をずらした。

また、試験観測の結果から昼、夜に一定値とさせている露出パラメーターも変更させるべきであることがわかった。最終的に、露出時間の昼の設定は 1/800s、夜の設定は 15s で、F 値は昼が f/8、夜は f/3.2 で、ISO 感度は昼が 200、夜は 1600 にした。

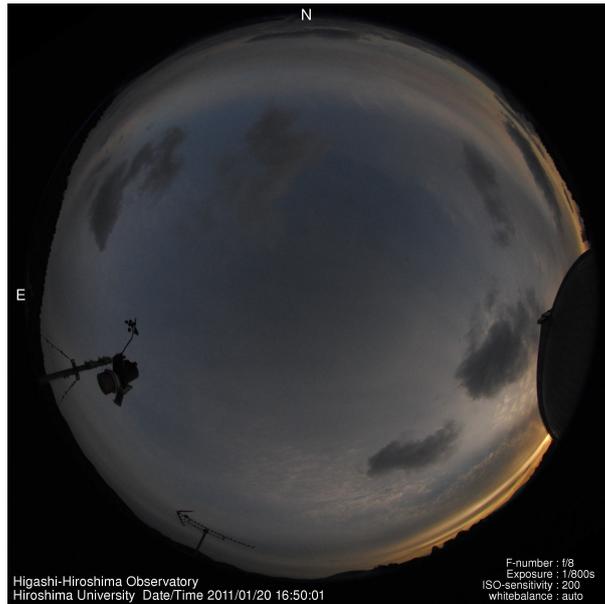
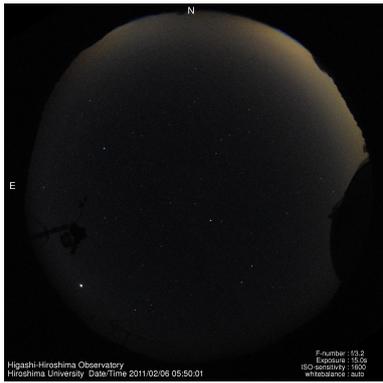
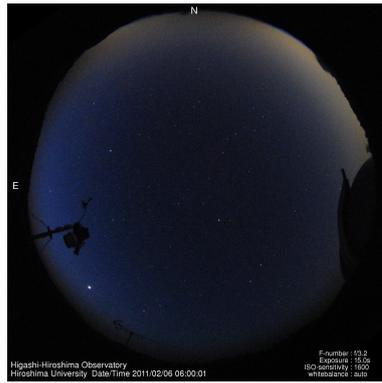


図 4.2: 日没時の計算プログラムを稼働させる前の時間の画像 (2011 年 1 月 20 日 16:50 SH=1/800 , F=f/8 , ISO=200)。全体的に暗く、雲の視認が難しい。

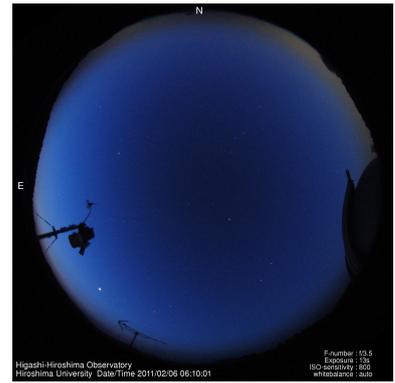
これらを含め、第 3 章の式 (3.3)~(3.8) に基いた計算プログラムで観測したものが次の図 4.3 である。



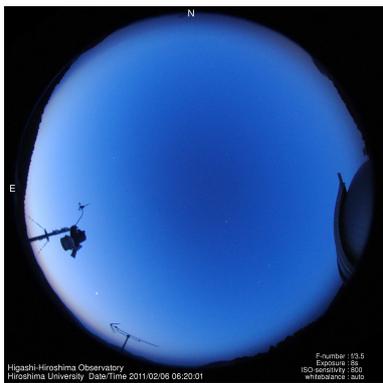
(a) 5:50 (SH=15 , F=f/3.2 , ISO=1600)



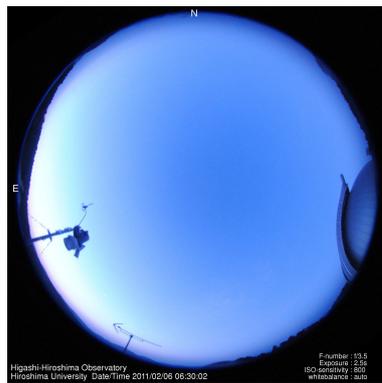
(b) 6:00 (SH=15 , F=f/3.2 , ISO=1600)



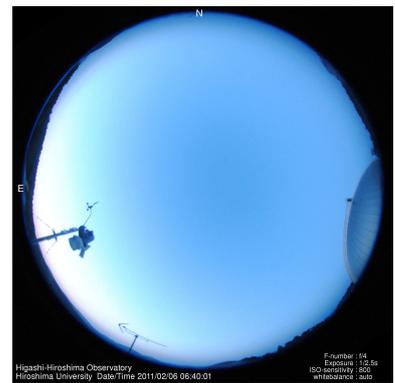
(c) 6:10 (SH=13 , F=f/3.2 , ISO=1600)



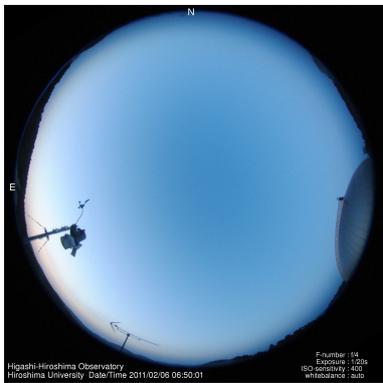
(d) 6:20 (SH=8 , F=f/3.5 , ISO=800)



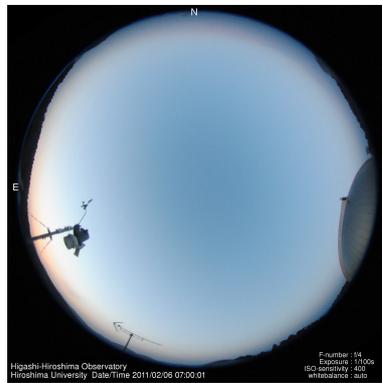
(e) 6:30 (SH=2.5 , F=f/3.5 , ISO=800)



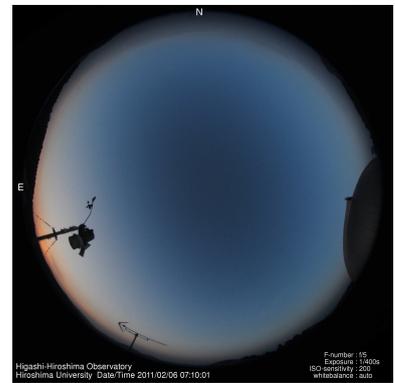
(f) 6:40 (SH=1/2.5 , F=f/4 , ISO=800)



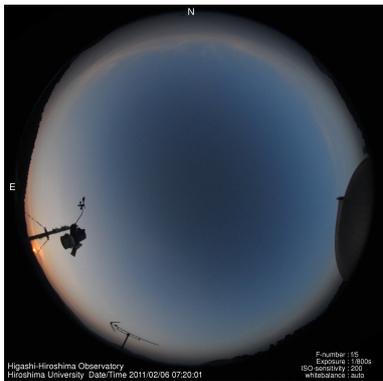
(g) 6:50 (SH=1/20 , F=f/20 , ISO=400)



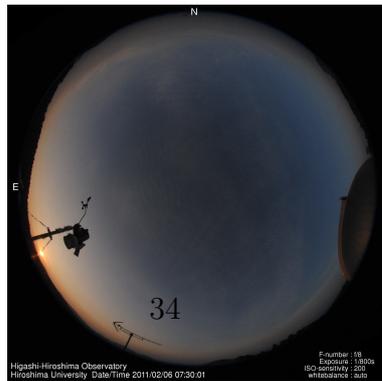
(h) 日の出 7:00 (SH=1/100 , F=f/4 , ISO=400)



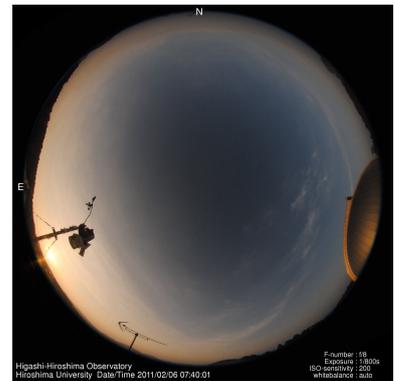
(i) 7:10 (SH=1/400 , F=f/5 , ISO=200)



(j) 7:20 (SH=1/800 , F=f/5 , ISO=200)



(k) 7:30 (SH=1/800 , F=f/8 , ISO=200)



(l) 7:40 (SH=1/800 , F=f/8 , ISO=200)

図 4.3: 改良後(2011年2月6日日の出)

図 4.3 では明るくなり過ぎて見えなくなる時間帯や、暗くて見にくくなる時間帯はなくなった。(a,b,c,d) では、明るい星もいくつか確認できた。また、(j,k,l) ではうっすらと雲が出てくる様子が確認できる。改良した計算プログラムの設定で日の出、日没時の雲の様子は目視で判別するのに十分な露出パラメーターでの撮影が可能となったと言える。

4.3 雲認知度合の評価

まず、観測に最も関わる夜の雲の視認度を見る。次の図 4.4 は 2011 年 2 月 2 日の夜 1:00~1:22 の 22 分間を 2 分間隔で撮った画像である。この時間帯は雲が全体にまだらに広がっており、雲の流れがよく見えた。

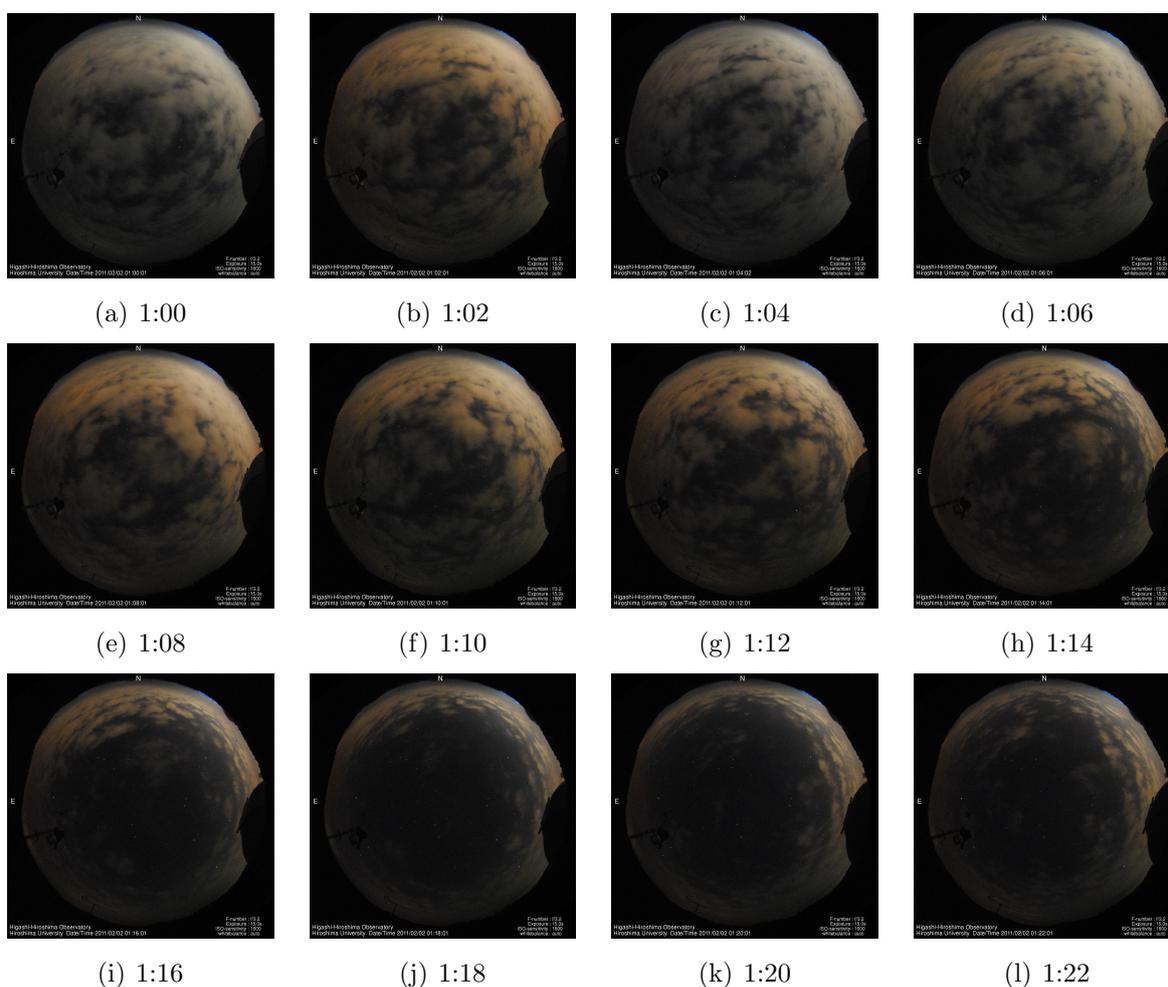
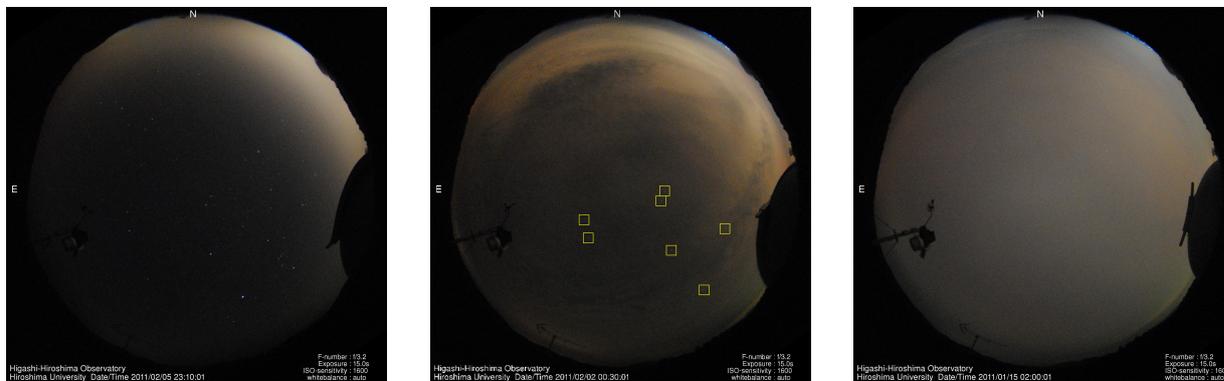


図 4.4: 夜間 (暗夜) の雲の流れの様子 (2011 年 2 月 2 日 SH=15 , F=f/3.2 , ISO=1600)。最初は全体にまだらに雲が広がっている様子が確認できる。時間を追って雲が流れていき、晴れていくのがわかる。また、全ての画像で雲の間に星が映っているのが確認できた。

図 4.4 からわかるように、最初の方の画像で空全体にまだらに広がる雲の様子が確認できる。時間を追う毎に雲が流れていき、だんだん晴れ間が広がっていく様子がわかる。また全ての時間で雲の間に星が確認でき、どこが晴れていて、どこが雲っているのか一目でわかる。



(a) 晴れ (2011 年 2 月 5 日 23:10 SH=15 , F=f/3.2 , ISO=1600) (b) 薄雲 (2011 年 2 月 2 日 0:30 SH=15 , F=f/3.2 , ISO=1600) (c) 曇天 (2011 年 1 月 15 日 2:00 SH=15 , F=f/3.2 , ISO=1600)

図 4.5: 左から雲のない日、薄雲の広がる日、厚い雲に覆われた日の画像。真ん中 (b) の画像は左 (a) と比べるとうっすらと雲が広がっているが、その下に星 (黄色四角) が映っているのがわかる。それに比べて右 (c) は全体に厚い雲で覆われている。

図 4.5 の 3 枚の画像は、全て夜のもので同じ露出パラメーターで撮影したものである。左 (a) と真ん中 (b) を比べると、真ん中 (b) はうっすらと雲が広がっているのがわかる。しかし、うっすら広がる雲の下、黄色の四角で囲んだ部分に星が映っているのがわかる。さらに、右 (c) の画像は全体的にぶ厚い雲がかかっており、星は確認できない。このことから、雲の有無だけでなく、雲のだいたいの厚さも目視で確認することができる。

図 4.4 や図 4.5 から、夜の時間帯で雲の形や流れる様子、さらに雲のだいたいの厚さを目視で確認するのに十分な画像を得られることがわかった。

図 4.6 は、2011 年 2 月 3 日の 12:30~12:44 の 14 分間の空の様子である。

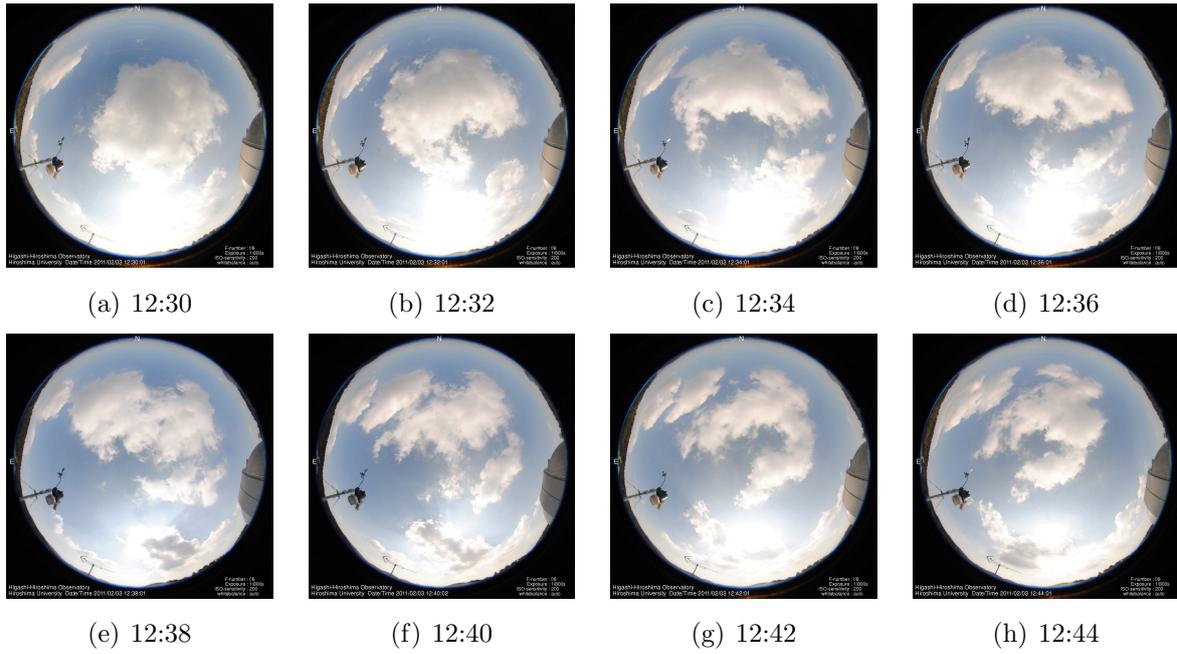


図 4.6: 昼間の雲の流る様子 (2011 年 2 月 3 日 SH=1/800 , F=f/8 , ISO=200)。雲の形がはっきりと映しだされており、雲の流れもよくわかる。



(a) 晴れ (2011 年 2 月 4 日 10:00 SH=1/800 , F=f/8 , ISO=200) (b) 薄雲 (2011 年 2 月 6 日 10:42 SH=1/800 , F=f/8 , ISO=200) (c) 曇天 (2011 年 2 月 2 日 14:44 SH=1/800 , F=f/8 , ISO=200)

図 4.7: 左から雲のない日、薄雲の広がる日、厚い雲に覆われた日の画像。左 (a) のように快晴の日でもサチュレートすることはない。また、真ん中 (b) や右 (c) のように雲の厚さもわかる。

図 4.6 や図 4.7 から、昼間は夜間より露出時間が短いので、その瞬間の雲の様子が映し出されていて、夜間よりも鮮明に雲の様子がわかる。また、図 4.7 から、天気によって露出パラメーターを変更しなくても、サチュレートを起こしたり、暗くて見えないというこ

とはなく、昼の露出パラメーターは、露出時間 1/800s、F 値は f/8、ISO 感度は 200 という固定の値を用いることで十分に雲の判別が可能である。

以上のことから、1日のどの時間帯においてもスカイモニターによる雲の視認が可能である。

4.4 月のフェーズによる雲の見栄え・恒星の限界等級

様々な月のフェーズをカバーする期間で試験観測を行い、月のフェーズ毎に雲の見栄え、限界等級を確認した。限界等級は視認可能な最も暗い星の等級と定義し、目視で判断した。

月のフェーズは新月期(月齢 0~3、26~29)、半月期(月齢 4~10、19~25)、満月期(月齢 11~18)に分けて考えた。

既存のスカイモニターで得られた限界等級は 3 等程度であったが、新しいスカイモニターでは、新月期、半月期で 5 等程度の恒星が確認できた。また、満月期においても、3 等程度の観測が可能であることがわかった。詳しくは次で述べる。

4.4.1 新月期(月齢 0~3、26~29)

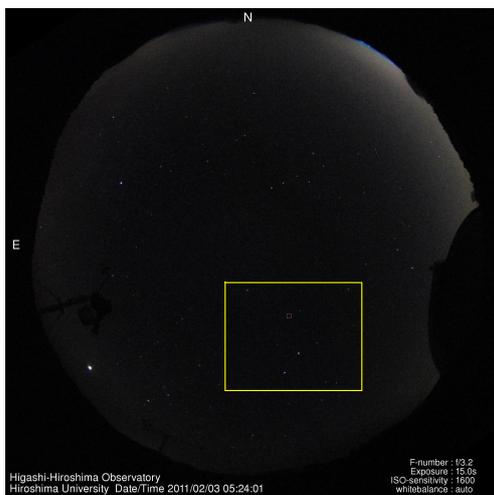


図 4.8: 新月期(2011年2月3日 5:24 月齢 0 SH=15, F=f/3.2, ISO=1600)の画像。



図 4.9: 図 4.8 の黄色い部分の拡大図。ピンクの四角で囲んだのが確認できた限界等級で 5 等星。

図 4.8、図 4.9 からわかるように、拡大することで 5 等星まで確認ができた。色調整した画像でさらに暗い星を見てみたところ、5 等星で拡大しただけでは見つらなかった星が見えるようになったが、6 等星までは確認できなかった。

新月期は設定している露出パラメーター (SH=15 , F=f/3.2 , ISO=1600) で、限界等級 5 等までの星が確認できた。これは第 2 章 1 節で見積った値である、4.5 等を上回る結果である。見積値を上回った理由としては、見積もりの際、スカイのノイズ以外は無視できるものと仮定したが、カメラの性能が向上し、1pix 当たりのノイズが軽減されたためだと考えられる。

この日 (2月3日) は天気がよく、雲は見られなかったが、次の図 4.10 に前日の同じ時間 (2011年2月2日 5:24 月齢 28.7) の画像を示す。

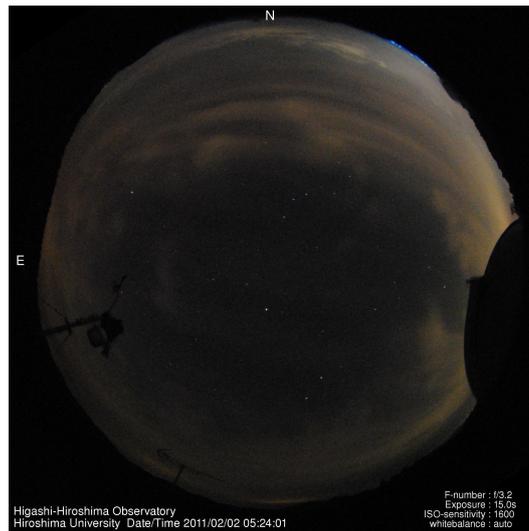


図 4.10: 新月期 (2011年2月4日 5:24 月齢 28.7 SH=15 , F=f/3.2 , ISO=1600) の画像。画像の上の方には帯状の雲が確認できる。薄雲の下には星も確認できる。

この図からわかるように、新月期の月がない状態においても、雲の形や厚さなどしっかりと確認することができ、設定している露出パラメーター (SH=15 , F=f/3.2 , ISO=1600) は雲の判別をするのに十分なものであるといえる。

4.4.2 半月期 (月齢 4~10、19~25)

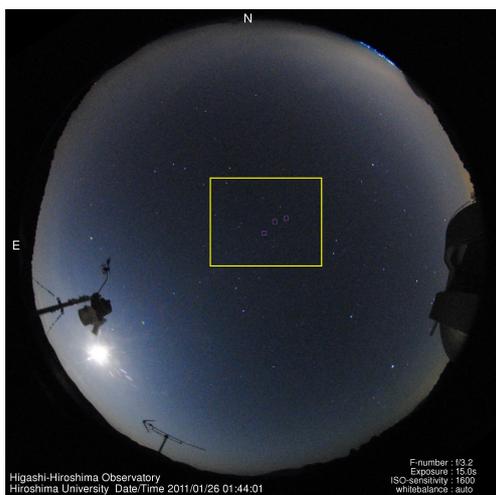


図 4.11: 半月 (2011 年 1 月 26 日 1:44 月齢 21.7 SH=15 , F=f/3.2 , ISO=1600) の画像。

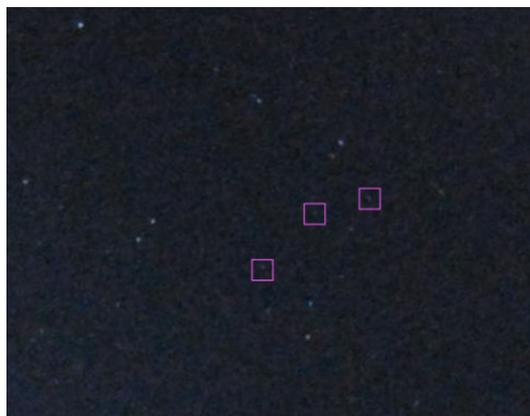


図 4.12: 図 4.11 の黄色い部分の拡大図。ピンクの四角で囲んだのが確認できた限界等級で 5 等星。

図 4.11、図 4.12 からわかるように、拡大して 5 等星が確認できた。新月期 (2011 年 2 月 3 日) と比べて天気がよかったため、たくさんの星が容易に確認できた。新月期より空が全体的に明るく映っており、5 等星がたくさんあるようにも見えた。しかし、中には星ではなくノイズであるものもあったので注意が必要である。

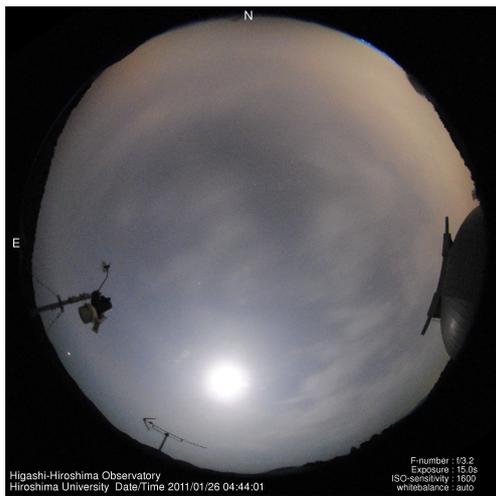


図 4.13: 半月期 (2011 年 1 月 26 日 4:44 月齢 21.7 SH=15 , F=f/3.2 , ISO=1600) の画像。全体に雲がかかっているが、月の位置が確認できる。また、雲の薄い部分では、星も見える。

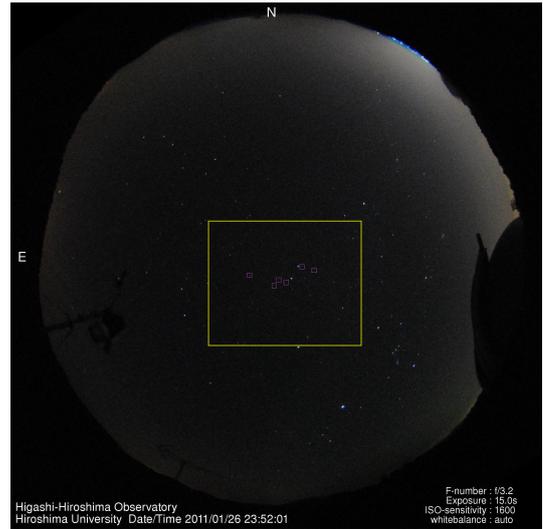


図 4.14: 半月期 (2011 年 1 月 26 日 23:52 月齢 22.7 SH=15 , F=f/3.2 , ISO=1600) の画像。黄色の四角で囲んだ部分を拡大して限界等級を調べた。天気がよく、新月期より多く 5 等星を確認することができた。

図 4.13 は同日 (2011 年 1 月 26 日) の 4:44 の画像である。全体に薄雲がかかっているが、月の位置がはっきりと確認でき、雲の薄い部分では星も確認できた。

図 4.14 は、半月期でも、月の上がっていない時間帯の画像である。新月期と同様の空が確認でき、この日は天気が良かったため、図 4.8 の 2011 年 2 月 3 日より、多くの星が確認できた。

以上のことから、半月期においても、新月期と同じパラメーター (SH=15 , F=f/3.2 , ISO=1600) で、限界等級 5 等星が確認でき、雲の判別も十分に行うことができるといえる。また、月の出ていない時間では、新月期同様に空の様子を見ることができる。

4.4.3 満月期 (月齢 11~18)

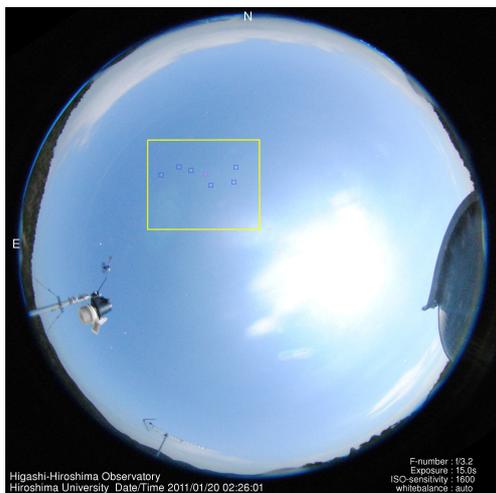


図 4.15: 満月期 (2011 年 1 月 20 日 2:26 月齢 15.7 SH=15 , F=f/3.2 , ISO=1600) の画像。

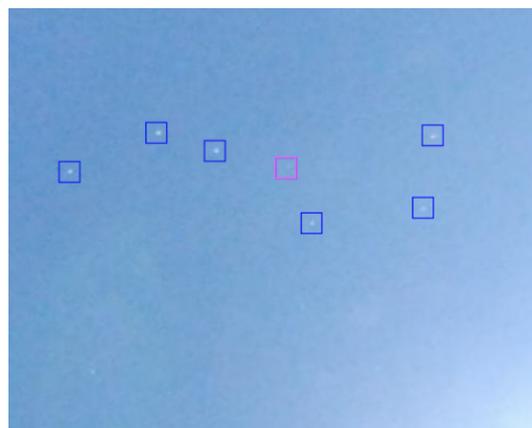


図 4.16: 図 4.15 の黄色い部分の拡大図。月が明るすぎて、見える星が少ない。青の四角で囲んだ北斗七星がやっと確認できる。限界等級はピンクの四角で囲んだ 3 等星。

図 4.15 からわかるように、月の散乱光の影響で空が明るくなりすぎていて、確認できるのは 3 等星程度である。この図の色レベルなどを調整したのが次の図 4.17 である。

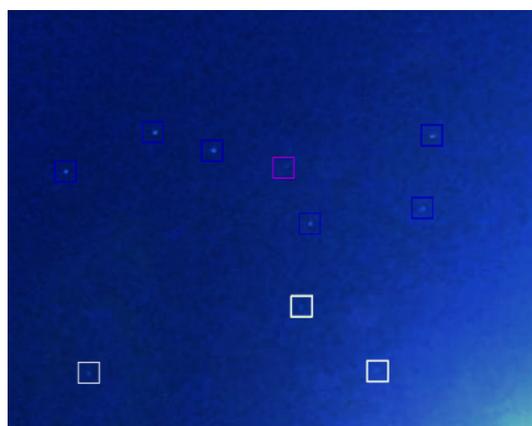


図 4.17: 図 4.14 と同じ画像の色レベルを調整した画像。月により近い星を確認することができたが、限界等級は 3 等星。

図 4.17 からわかるように、色レベルの調整を行うことで、より月に近くにある星が確認できるようになったが、視認できたのは 3 等星どまりで、4 等星は見えなかった。

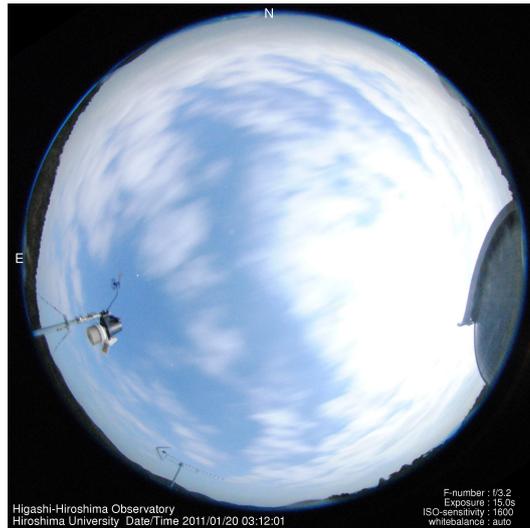


図 4.18: 満月期 (2011 年 1 月 20 日 3:12 月齢 15.7 SH=15 , F=f/3.2 , ISO=1600) の画像。全体にまだらに雲があるのが確認できる。露出時間の長さの分、雲の像が伸びて見えることもわかる。よって雲の形の判別は可能である。

満月期で雲のある時間帯の画像が図 4.18 である。この画像から、設定している露出パラメーター (SH=15 , F=f/3.2 , ISO=1600) で、雲の判別は満月期も十分に可能であるといえる。

以上のことから、満月期は新月期や半月期と同じ露出パラメーター (SH=15 , F=f/3.2 , ISO=1600) で撮影を行うと、月の明るさから空が明るく映ってしまい、限界等級が 3 等になってしまうものの、雲の判別には特に問題がないことがわかった。

空の明るさを抑えるために、満月期の露出パラメーターを変更することも検討したが、恒星の限界等級が下ってしまうことや、雲が濃い日には露出が足りず、雲の判定ができなくなる懸念されるため、同じ露出パラメーターのまま充分であると判断した。

第5章 まとめと今後

私は、広島大学宇宙科学センター附属東広島天文台のかなた望遠鏡での、より効率的な観測へ向けた雲の検知と、予測不能な突発的天体の検知を目標に、スカイモニターのアップグレードを行った。また、さらなる効率的な観測と、将来的に大学キャンパスからの遠隔操作によって観測を行うことを可能にするため、観測環境情報の統合化と異常時には警告を表示させるシステムの開発を行った。

まず、既存のスカイモニターで使用しているビデオカメラを、より分解能が高く、遠隔操作によるPCからの制御が可能なデジタルカメラへと交換した。既存のスカイモニターで使用していたカメラ以外の筐体は引き継いで使い、カメラを固定する治具に必要な加工を施した。

次に、時間(太陽高度)に合わせて最適な露出パラメーターを計算するプログラムを作成し、そのプログラムによって求めたパラメーターでの画像の取得、加工、webに更新、アーカイブに保存までを自動で行うソフトウェアの開発を行った。数日間の試験観測を行い、時間に合わせて1日中サチュレートを起こすことのない露出パラメーターを設定し、撮影をし続けるプログラムを作成することができた。得られた画像では、どの月のフェーズでも、問題なく雲の視認ができた。また、天気の良い新月期から半月期の画像で、5等程度の恒星まで視認ができた。これによって、既存のスカイモニターでは撮影することのできなかつた時間帯(昼間、薄明時、満月時)の撮影が可能となり、限界等級も2等程度向上させることができた。

さらに、今まで別々に確認しなければならなかつた観測環境情報の統合化を一部であるが行うことができた。図5.1は統合化させた情報を掲載する予定のwebサイトである。観測環境情報(スカイモニター、気象センサー、雨滴センサーの情報)だけでなく、地方恒星時、望遠鏡やドームの情報(向き等)も一度に確認できるようにした。現在、気象センサーや望遠鏡、ドームの情報を自動で更新させることのできるプログラムを作成している。そのため、まだ実用化させてはいないが、今後、非常時に警告を表示するシステムを作成し、でき次第実用化させる予定である。これが実用化されることで、かなた望遠鏡による観測がより効率的なものとなり、将来、大学キャンパスからの遠隔操作による観測も可能となると期待できる。

ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H)

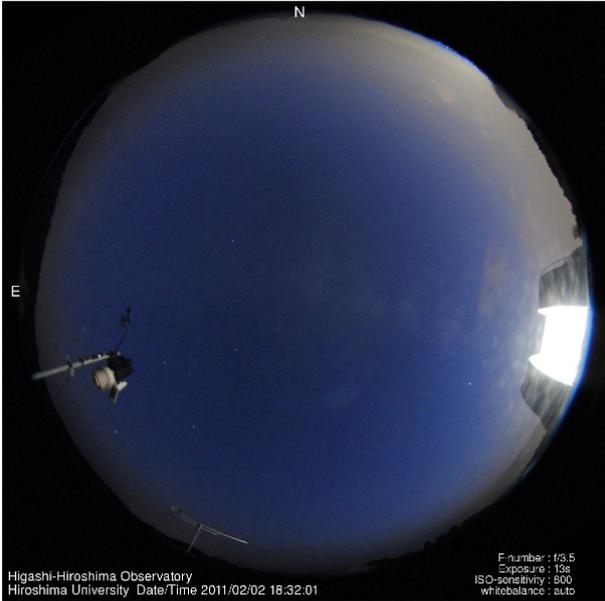
http://localhost/index3.html

よく見るページ オンラインマニュアル Yahoo! JAPAN Higashi-Hiroshima O... Gmail - 受信トレイ (... 地球人ネットワークを... Current Weather at ...

Gmail - 【お願い】 ~給与、旅... OAO Skymonitor Higashi-Hiroshima Observat... Current Weather at Higashi-...

Current Weather at Higashi-Hiroshima Observatory

スカイモニター



現在時
2011/02/09 (Wed) 21:46:28

現在の地方恒星時
01:10:47.00

気象センサー					
	Temperature (°C)	Humidity (%)	Wind Direction	Wind Speed(m/s)	Rainfall (mm)
Outside	19.9	54	NNW	0.9	0.00
Dome Inside	21.4	35	-	-	-

雨滴センサー
ON OFF

望遠鏡			
Control Device	Equatorial Coordinates	Horizontal Coordinates	GRB mode
ON	Ra = 01:10:38.20 Dec = +33:23:51.1	Az = +3.0000 Alt = +89.0000	ON

ドーム

DomeSlit	MirrorCover	向き
Open	Open	4.2 (S)

リンク

- [Ephemeris 今日のこよみ](#)
- [天文台監視カメラ](#)

完了

図 5.1: 開発途中の観測環境情報を統合化した web サイト。

付録 A ソースコード

付録 A.1 画像の自動取得・加工・更新プログラム

```
#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <ctype.h>
#include "Shutter.h"
#include "Fnumber.h"
#include "Iso.h"

/* HHAO */
double longitude = +132.7767;
double latitude = +34.3775;
double altitude = +511.2;

char str1[25], str2[25], str3[25], str4[25];
double ONEDEG = 0.017453292;
char fn[30];
char sp[30];
char iso[30];
char wh[30];

char *mid( char *, int, int );
char *convstr(char *);
char *fitsdate(const struct tm *);
char *fitstime(const struct tm *);
char *fitsdatemjd(const double);
char *fitstimemjd(const double);
char *fitsra(const double);
char *fitsdec(const double);
double timetomjd(int, int, int, int, int);
double getmjd(tm *);
void mjdtoime(double, int *, int *, int *, int *,
int *, int *);
double getst(double, double);
double getha(double, double);
double getzd(double, double, double);

double spls( double ); /* solar position1
(celestial longitude, degree) */
double spds( double ); /* solar position2
(distance, AU) */
double spal( double ); /* solar position3
(declination, degree) */
double spdl( double ); /* solar position4
(the right ascension, degree) */
double sh( double, int, int, int, double, double);
/* Calculate sidereal hour (degree) */
double eandp( double, double); /* Calculating the
seeming horizon altitude "sa"(degree) */
double sa( double, double);
double soal( double, double, double, double ); /*
Calculating solar alititude (degree) */
double sodr( double, double, double, double ); /*
Calculating solar direction (degree) */

double hho_sunrise_today( void );
double hho_sunset_today( void );

char *mid(char *string, int n1, int n2 ){
int i;
for( i=0; i<=(n2-1); i++ )
str1[i] = string[n1+i-1];
str1[n2] = '\0';
return( str1 );
}

/****** CONVERSION OF STRING *****/
char *convstr(char *string){
sprintf( str1, "%s", string );
return( str1 );

/***** CONVERSION from CONSTRUCT tm TYPE VARIABLE to
FITS STRING *****/
char *fitsdate(const struct tm *tmptime){
int tmpyear;
/* before 1999 */
sprintf(str2, "%02d/%02d/%02d", tmptime->tm_mday,
(1 + tmptime->tm_mon), tmptime->tm_year);
/*
**** after 2000 ****/
if( tmptime->tm_year > 70 )
tmpyear = 1900 + tmptime->tm_year;
else
tmpyear = 2000 + tmptime->tm_year;
sprintf(str2, "%04d-%02d-%02d", tmpyear, (1 + tmptime
->tm_mon), tmptime->tm_mday );
/**/
return( str2 );
}

char *fitstime(const struct tm *tmptime){
sprintf(str3, "%02d:%02d:%02d", tmptime->tm_hour,
tmptime->tm_min, tmptime->tm_sec);
return(str3);
}

/***** CONVERSION from MJD(double precision) to FITS
STRING *****/
char *fitsdatemjd(const double mjd){
int year, month, day, hour, min, sec;

mjdtoime(mjd, &year, &month, &day, &hour, &min,
&sec);

/*
if(year < 2000){
year -= 1900;
sprintf(str2, "%02d/%02d/%02d", day, month, year);
}
else{
sprintf(str2, "%4d-%02d-%02d", year, month, day);
}
*/
sprintf(str2, "%04d-%02d-%02d", year, month, day);

return(str2);
}

char *fitstimemjd(const double mjd){
int year, month, day, hour, min, sec;

mjdtoime(mjd, &year, &month, &day, &hour, &min, &sec);
sprintf(str3, "%02d:%02d:%02d", hour, min, sec);
return(str3);
}

/***** RA (rad) -> FITS STRING *****/
char *fitsra(const double a){
double b;
int h, m, s;

b = a*57.2957795130824/15.0;
h = b + 0.000001; /* avoid numerical-round (?) */
b = (b - h) * 60.0;
m = b + 0.000001;
b = (b - m) * 60.0;
s = b + 0.000001;
sprintf(str2, "%02d:%02d:%02d", h, m, s);
return(str2);
}

/***** DEC (RAD) -> FITS STRING *****/
char *fitsdec(const double a){
double b;
int d, m, s;
char sn='+';

b = a;
if(b < 0.0){
sn='-';
b = fabs(b);
}
b = b*57.2957795130824;
d = b + 0.000001; /* avoid numerical-round (?) */
```

```

b = (b - d) * 60.0;
m = b + 0.000001;
b = (b - m) * 60.0;
s = b + 0.000001;
sprintf(str3, "%c%02d:%02d:%02d", sn, d, m, s);
return(str3);
}

/***** CONVERSION from DATE, TIME to MJD
(= JD - 2400000.5) *****/
double timetomjd(int year, int month, int day,
int hour, int min, int sec){
double mjd;
long i1, i2, i3, i4;

if(month <= 2) {year--; month+=12;}
i1 = 365.25*year;
i2 = year/400.0;
i3 = year/100.0;
i4 = 30.59*(month-2.0);
mjd = i1 + i2 - i3 + i4 + day - 678912;
mjd += (double)hour/24.0 + (double)min/1440.0
+ (double)sec/86400.0;

return(mjd);
}

/***** 2000.0年からの経過時間(単位年) *****/
double ey2000(double mjd){
double mjd2000;
mjd2000 = timetomjd(2000, 1, 1, 0, 0, 0);
return((mjd - mjd2000) / 365.25);
}

/***** tm構造体から直接MJDを得るときのバイパス関数
*****/
double getmjdtm(struct tm *tmptime){
double mjd;
if(tmptime->tm_year > 70)
mjd = timetomjd((1900 + tmptime->tm_year), (1
+tmptime->tm_mon), tmptime->tm_mday, tmptime->
tm_hour, tmptime->tm_min, tmptime->tm_sec);
else
mjd = timetomjd((2000 + tmptime->tm_year), (1
+tmptime->tm_mon), tmptime->tm_mday, tmptime->
tm_hour, tmptime->tm_min, tmptime->tm_sec);
return(mjd);
}

/***** CONVERSION from MJD(= JD - 2400000.5) to DATE,
TIME *****/
void mjdtoime(double mjd, int *year, int *month,
int *day, int *hour, int *min, int *sec){
int i, n;
int ymonth[]={0, 31, 28, 31, 30, 31, 30, 31, 31, 30,
31, 30, 31};
double submjd, subday;
double sp1, sp2;
double a1=0.0027379093;

*year = a1 * mjd + 1858.877;
*month = 1;
*day = 0;
submjd = timetomjd(*year, *month, *day, 0, 0, 0);
sp1 = mjd - submjd;
if((*year % 4) == 0) ymonth[2] = 29;
if((*year % 100) == 0 && (*year % 400) != 0)
ymonth[2]=28;
sp2 = 0;
for(i=1; i<=12; i++){
n = sp1;
if((n - sp2 - ymonth[i]) <= 0) break;
sp2 += ymonth[i];
}
*month = i;
subday = sp1 - sp2;
if(*month >= 13) {*year += 1; *month -= 12;}
if(subday < 1) {*year -= 1; *month = 12;}
subday += 31;
*day = subday;
subday -= *day;
subday *= 24.0;
*hour = subday;
subday -= *hour;
subday *= 60.0;
*min = subday;
subday -= *min;
subday *= 60.0;
*sec = subday;
}

subday *= 60.0;
*sec = subday;
}

/***** CALCULATION OF SIDEREAL TIME(rad) *****/
/* mjd: MJD(UT), lo: LONGITUDE of TELESCOPE
(rad 東経+ 西経-) */
double getst(double mjd, double lo){
double st;
long i;
double a1=0.671262;
double a2=1.002737909;

st = a1 + a2*(mjd - 40000.0) + lo/6.283185307;
i = st;
st = 6.283185307*(st - i);
return(st);
}

/***** CALCULATION OF HOUR ANGLE(rad) *****/
/* a: SIDEREAL TIME(rad), slo: RA(rad) of the STAR */
double getha(double a, double slo){
double ha;
ha = a - slo;
return(ha);
}

/***** CALCULATION OF ZENITH DISTANCE(rad) *****/
/* d:DEC(rad) of the STAR, phi: LATTITUDE of
TELESCOPE(rad 北緯+ 南緯-),
ha: HOUR ANGLE of the star, h: HIGHT(rad) */
double getzd(double d, double phi, double ha){
double zd, h;
h = asin(sin(d)*sin(phi) + cos(d)*cos(phi)*cos(ha));
zd = 1.570796327 - h;
return(zd);
}

/**** solar position1 (celestial longitude, degree) ****/
double spls(double t) { // t: Julius year-jd2000
double l;
l = 280.4603 + 360.00769 * t \
+ (1.9146 - 0.00005 * t) * sin( ( 357.538 + 359.991
* t ) * ONEDEG)\
+ 0.0200 * sin( ( 355.05 + 719.981 * t ) * ONEDEG)\
+ 0.0048 * sin( ( 234.95 + 19.341 * t ) * ONEDEG)\
+ 0.0020 * sin( ( 247.1 + 329.640 * t ) * ONEDEG)\
+ 0.0018 * sin( ( 297.8 + 4452.67 * t ) * ONEDEG)\
+ 0.0018 * sin( ( 251.3 + 0.20 * t ) * ONEDEG)\
+ 0.0015 * sin( ( 343.2 + 450.37 * t ) * ONEDEG)\
+ 0.0013 * sin( ( 81.4 + 225.18 * t ) * ONEDEG)\
+ 0.0008 * sin( ( 132.5 + 659.29 * t ) * ONEDEG)\
+ 0.0007 * sin( ( 153.3 + 90.38 * t ) * ONEDEG)\
+ 0.0007 * sin( ( 206.8 + 30.35 * t ) * ONEDEG)\
+ 0.0006 * sin( ( 29.8 + 337.18 * t ) * ONEDEG)\
+ 0.0005 * sin( ( 207.4 + 1.50 * t ) * ONEDEG)\
+ 0.0005 * sin( ( 291.2 + 22.81 * t ) * ONEDEG)\
+ 0.0004 * sin( ( 234.9 + 315.56 * t ) * ONEDEG)\
+ 0.0004 * sin( ( 157.3 + 299.30 * t ) * ONEDEG)\
+ 0.0004 * sin( ( 21.1 + 720.02 * t ) * ONEDEG)\
+ 0.0003 * sin( ( 352.5 + 1079.97 * t ) * ONEDEG)\
+ 0.0003 * sin( ( 329.7 + 44.43 * t ) * ONEDEG);
while(l >= 360)
l -= 360;
while(l < 0)
l += 360;
return( l );
}

/**** solar position2 (distance, AU) ****/
double spds(double t) { // t: Julius year-jd2000
double r;
r = (0.007256 - 0.0000002 * t) * sin( ( 267.54 + 359.991
* t ) * ONEDEG)\
+ 0.000091 * sin( ( 265.1 + 719.98 * t ) * ONEDEG)\
+ 0.000030 * sin( ( 90.0 ) * ONEDEG)\
+ 0.000013 * sin( ( 27.8 + 4452.67 * t ) * ONEDEG)\
+ 0.000007 * sin( ( 254 + 450.4 * t ) * ONEDEG)\
+ 0.000007 * sin( ( 156 + 329.6 * t ) * ONEDEG );
r = pow( 10.0, r );
return( r );
}

/**** solar position3 (declination, degree) ****/
double spls(double t) { // t: Julius year-jd2000
double ls, ep, al;
ls = spls( t );
}

```

```

    ep = 23.439291 - 0.000130042 * t;
    al = atan( tan( ls * ONEDEG ) * cos( ep *
ONEDEG ) ) / ONEDEG;
    if( ( ls >= 0. ) && ( ls < 180. ) ){
        while(al < 0.)
            al += 180.;
        while(al >= 180.)
            al -= 180.;
    }
    else{
        while(al < 180.)
            al += 180.;
        while(al >= 360.)
            al -= 180.;
    }
    return( al );
}

/** solar position4 (the right ascension, degree)
***/
double spd1( double t ) { // t: Julius year-jd2000
    double ls, ep, dl;
    ls = spls( t );
    ep = 23.439291 - 0.000130042 * t;
    dl = asin( sin( ls * ONEDEG ) * sin( ep * ONEDEG
) ) / ONEDEG;
    return( dl );
}

/** Calculate sidereal hour (degree) ***/
double sh( double t, int h, int m, int s, double l,
double i ) {
    // t: julius year-jd2000, h: hour, m: minute, s:
second,
    // l: longitude, i: time difference
    double d, th;
    d = (( s / 60. + m ) / 60. + h ) / 24. ; // elapsed
hour (from 0:00 a.m.)
    th = 100.4606 + 360.007700536 * t + 0.00000003879
* t * t - 15. * i ;
    th += l + 360 * d ;
    while(th >= 360.)
        th -= 360.;
    while(th < 0.)
        th += 360.;
    return( th );
}

/** Calculating the seeming horizon altitude
"sa"(degree) ***/
double eandp( double alt, double ds ) { //
subfunction for altitude and parallax
    double e, p;
    e = 0.035333333 * sqrt( alt ) ;
    p = 0.002442818 / ds ;
    return( p - e );
}

double sa( double alt, double ds ) { // alt:
altitude (m), ds: solar distance (AU)
    double s, r, k;
    s = 0.266994444 / ds ;
    r = 0.585555555 ;
    k = eandp( alt, ds ) - s - r ;
    return( k );
}

/** Calculating solar alititude (degree) ***/
double soal( double la, double th, double al,
double dl ) {
    // la: latitude, th: sidereal hour,
    // al: solar declination, dl: right ascension
    double h;
    h = sin( dl * ONEDEG ) * sin( la * ONEDEG ) \
+ cos( dl * ONEDEG ) * cos( la * ONEDEG ) *
cos( (th - al) * ONEDEG ) ;
    h = asin( h ) / ONEDEG;
    return( h );
}

/** Calculating solar direction (degree) ***/
double sodr( double la, double th, double al,
double dl ) {
    // la: latitude, th: sidereal hour,
    // al: solar declination, dl: right ascension
    double t, dc, dm, st, dr;
    t = th - al ;
    dc = - cos( dl * ONEDEG ) * sin( t * ONEDEG );
    dm = sin( dl * ONEDEG ) * sin( la * ONEDEG ) \
- cos( dl * ONEDEG ) * cos( la * ONEDEG ) *
cos( t * ONEDEG ) ;
    if(dm == 0){
        st = sin( t * ONEDEG );
        if(st > 0.) dr = -90. ;
        if(st == 0) dr = 9999. ;
        if(st < 0) dr = 90. ;
    }
    else {
        dr = atan(dc / dm) / ONEDEG;
        if(dm < 0.) dr += 180. ;
    }
    if(dr < 0.) dr += 360 ;
    return( dr );
}

double hho_sunrise_today( void ){
    time_t t;
    struct tm *tp;
    // char buf[200];
    double ut;
    int year, mo, da, ho, mi, se;
    // double al;
    int hh, m;
    double tjd, th, ds, ls, alp, dlt, pht, ht, t4;
    double tsec_today;

    t = time(NULL);
    tp = localtime( &t );

    da = tp->tm_mday;
    mo = tp->tm_mon+1;
    year = tp->tm_year;
    if( year > 70 )
        year += 1900;
    else
        year += 2000;
    ho = tp->tm_hour;
    mi = tp->tm_min;
    se = tp->tm_sec;

    ut = timetomjd( year, mo, da, ho, mi, se ) -
0.375;
    tjd = ey2000( timetomjd( year, mo, da-1, 23,
59, 0 ) );
    th = sh( tjd, 23, 59, 0, longitude, +9. ) ;
    ds = spds( tjd ) ;
    ls = spls( tjd ) ;
    alp = spal( tjd ) ;
    dlt = spd1( tjd ) ;
    pht = soal( latitude, th, alp, dlt ) ;

    for( hh=0; hh<24; hh++){
        for( m=0; m<60; m++){
            tjd = ey2000( timetomjd( year, mo, da, ho,
mi, se ) );
            th = sh( tjd, hh, m, 0, longitude, +9. ) ;
            ds = spds( tjd ) ;
            ls = spls( tjd ) ;
            alp = spal( tjd ) ;
            dlt = spd1( tjd ) ;
            ht = soal( latitude, th, alp, dlt ) ;
            t4 = sa( altitude, ds ) ;

            if( ( pht < t4 ) && ( ht > t4 ) ){
                // printf( "Sunrise.%02d:%02d", hh, m );
                tsec_today = hh * 3600. + m * 60.;
            }
        }
        pht = ht;
    }
    return( tsec_today );
}

double hho_sunset_today( void ){
    time_t t;
    struct tm *tp;
    // char buf[200];
    double ut;
    int year, mo, da, ho, mi, se;
    // double al;
    int hh, m;
    double tjd, th, ds, ls, alp, dlt, pht, ht, t4;
    double tsec_today;

    t = time(NULL);
    tp = localtime( &t );

    da = tp->tm_mday;

```

```

mo = tp->tm_mon+1;
year = tp->tm_year;
if( year > 70 )
    year += 1900;
else
    year += 2000;
ho = tp->tm_hour;
mi = tp->tm_min;
se = tp->tm_sec;

ut = timetomjd( year, mo, da, ho, mi, se ) -
0.375;
tjd = ey2000( timetomjd( year, mo, da-1, 23,
59, 0 ) );
th = sh( tjd, 23, 59, 0, longitude, +9. ) ;
ds = spds( tjd ) ;
ls = spls( tjd ) ;
alp = spal( tjd ) ;
dlt = spdl( tjd ) ;
pht = soal( latitude, th, alp, dlt ) ;

for( hh=0; hh<24; hh++){
    for( m=0; m<60; m++){
        tjd = ey2000( timetomjd( year, mo, da,
ho, mi, se ) );
        th = sh( tjd, hh, m, 0, longitude, +9. ) ;
        ds = spds( tjd ) ;
        ls = spls( tjd ) ;
        alp = spal( tjd ) ;
        dlt = spdl( tjd ) ;
        ht = soal( latitude, th, alp, dlt ) ;
        t4 = sa( altitude, ds ) ;

        if( ( pht > t4 ) && ( ht < t4 ) ){
            // printf( " ,Sunset,%02d:%02d", hh, m ) ;
            tsec_today = hh * 3600. + m * 60.;
        }

        pht = ht;
    }
}
return( tsec_today );
}

```

```

/*
tjd = ey2000( timetomjd( year, mo, da-1, 23,
59, 0 ) );
th = sh( tjd, 23, 59, 0, longitude, +9. ) ;
ds = spds( tjd ) ;
ls = spls( tjd ) ;
alp = spal( tjd ) ;
dlt = spdl( tjd ) ;
pht = soal( latitude, th, alp, dlt ) ;
pdr = sodr( latitude, th, alp, dlt ) ;

for( hh=0; hh<24; hh++){
    for( m=0; m<60; m++){
        tjd = ey2000( timetomjd( year, mo, da,
ho, mi, se ) );
        th = sh( tjd, hh, m, 0, longitude, +9. ) ;
        ds = spds( tjd ) ;
        ls = spls( tjd ) ;
        alp = spal( tjd ) ;
        dlt = spdl( tjd ) ;
        ht = soal( latitude, th, alp, dlt ) ;
        dr = sodr( latitude, th, alp, dlt ) ;
        tt = eandp( altitude, ds ) ;
        t1 = tt - 18. ;
        t2 = tt - 12. ;
        t3 = tt - 6. ;
        t4 = sa( altitude, ds ) ;

        if( ( pht < t1 ) && ( ht > t1 ) )
            printf( " ,Twilight_start,%02d:%02d",
hh, m );
        if( ( pht < t4 ) && ( ht > t4 ) )
            printf( " ,Sunrise,%02d:%02d", hh, m );
        if( ( pdr < 180. ) && ( dr > 180. ) )
            printf( " ,Culmination,%02d:%02d", hh,
m );
        if( ( pht > t4 ) && ( ht < t4 ) )
            printf( " ,Sunset,%02d:%02d", hh, m );
        if( ( pht > t1 ) && ( ht < t1 ) )
            printf( " ,Twilight_end,%02d:%02d",
hh, m );
        pht = ht;
        pdr = dr;
    }
}

```

```

}
}
*/

int main(void)
{
    time_t timer;
    struct tm *t_st;
    int h , min , sec;
    double a , b ;
    int now , sh , f , is ;
    char cmd1[100];
    char cmd2[100];
    char cmd3[100];
    char cmd4[100];
    char cmd5[100];
    char cmd6[1000];
    char cmd7[1000];
    char cmd8[1000];
    char cmd9[1000];
    char cmd10[1000];
    char cmd11[1000];
    char cmd12[1000];
    char cmd13[1000];
    char cmd14[1000];
    char cmd15[1000];
    char cmd16[1000];
    char cmd17[1000];
    char cmd18[1000];
    char name1[20];
    char name2[30];
    char name3[30];
    char name4[30];
    char name5[30];
    char name6[30];
    char name7[30];
    char name8[30];
    char name9[30];
    int smok;
    int smo;

```

```

/*設定*/
sprintf( wh , "auto" ); /*whitebalance*/
/* F-number : fnumber.c で変更
shutterspeed : shutterspeed.c で変更
ISO : iso.c で変更
*/

/*現在の時刻*/
time(&timer);
printf( "現在の時刻:%s\n" , ctime(&timer));

t_st = localtime(&timer);
h = t_st->tm_hour;
min = t_st->tm_min;
sec = t_st->tm_sec;

now = h*60*60+min*60+sec;

printf( "%d\n" , now );

/*sunrise & sunset */
printf( "Today Sunrise: %5.0f sec Sunset: %5.0f
sec\n",\
        hho_sunrise_today(), hho_sunset_today() );
a = hho_sunrise_today();
b = hho_sunset_today();
printf( "%f,%f\n" , a , b );

/*shutterseed*/
sh = shutterspeed( a , b , now );

/*f-number*/
f = fnumber( a , b , now );

/*ISO*/

```

```

is = sensitivity( a , b , now );

/*日時*/
sprintf( name1, "%04d%02d%02d%02d%02d" ,
t_st->tm_year+1900,t_st->tm_mon+1,t_st->tm_mday,
t_st->tm_hour,t_st->tm_min,t_st->tm_sec);
sprintf( name2, "%04d" , t_st->tm_year+1900 );
sprintf( name3, "%04d%02d" , t_st->tm_year+1900,
t_st->tm_mon+1 );
sprintf( name4,"%04d%02d%02d" , t_st->tm_year+1900,
t_st->tm_mon+1,t_st->tm_mday );
sprintf( name5, "%04d/%02d/%02d %02d:%02d:%02d" ,
t_st->tm_year+1900,t_st->tm_mon+1,t_st->tm_mday,
t_st->tm_hour,t_st->tm_min,t_st->tm_sec );
sprintf( name6, "%02d%02d" , t_st->tm_hour,t_st->
tm_min );
sprintf( name7, "%04d%02d%02d%02d%02d" , t_st->
tm_year+1900,t_st->tm_mon+1,t_st->tm_mday,t_st->
tm.hour,t_st->tm.min);

/*写真を撮る*/
sprintf( cmd3 ,"/usr/local/bin/gphoto2 --quiet
--capture-image-and-download --filename
/iraf/iraf/local/skymon/photoprogram2/
current_master.jpg\n" );
system( cmd3 );

/*緑を付ける*/
sprintf( cmd17 , "/usr/bin/convert -border
800x800 -bordercolor black /iraf/iraf/local/
skymon/photoprogram2/current_master.jpg
/iraf/iraf/local/skymon/photoprogram2/
current_border.jpg" );
system( cmd17 );

/*回転させる*/
sprintf( cmd5 , "/usr/bin/convert -rotate 146
/iraf/iraf/local/skymon/photoprogram2/
current_border.jpg /iraf/iraf/local/skymon
/photoprogram2/current_rotate.jpg\n" );
system( cmd5 );

/*切り取る*/
sprintf( cmd4 , "/usr/bin/convert -crop
2200x2200+2370+2190 /iraf/iraf/local/skymon
/photoprogram2/current_rotate.jpg /iraf/iraf
/local/skymon/photoprogram2/current_crop.jpg
\n" );
system( cmd4 );

/*文字を入力する*/
/*Higashi-Hiroshima Observatory*/
sprintf( cmd10 , "/usr/bin/convert -font
helvetica -fill white -pointsize 48 -draw
'text 20,2120 \ "Higashi-Hiroshima Observatory\"
/iraf/iraf/local/skymon/photoprogram2/
current_crop.jpg /iraf/iraf/local/skymon
/photoprogram2/current_font.jpg" );
system( cmd10 );

/*Hiroshima University Date/Time*/
sprintf( cmd11 , "/usr/bin/convert -font
helvetica -fill white -pointsize 48 -draw
'text 20,2170 \ "Hiroshima University Date/Time
%s\"' /iraf/iraf/local/skymon/photoprogram2
/current_font.jpg /iraf/iraf/local/skymon/
photoprogram2/current_font.jpg" ,name5 );
system( cmd11 );

/* N */
sprintf( cmd7 , "/usr/bin/convert -font
helvetica -fill white -pointsize 56 -draw
'text 1060,60 N' /iraf/iraf/local/skymon/
photoprogram2/current_font.jpg /iraf/iraf
/local/skymon/photoprogram2/current_font.jpg
\n" );
system( cmd7 );

/* E */
sprintf( cmd7 , "/usr/bin/convert -font
helvetica -fill white -pointsize 56 -draw
'text 30,1080 E' /iraf/iraf/local/skymon/
photoprogram2/current_font.jpg /iraf/iraf

/local/skymon/photoprogram2/current_font.jpg
\n" );
system( cmd7 );

/*F-number*/
sprintf( cmd12 , "/usr/bin/convert -font
helvetica -fill white -pointsize 40 -draw
'text 1800,2050 \ "F-number : %s\"' /iraf/iraf
/local/skymon/photoprogram2/current_font.jpg
/iraf/iraf/local/skymon/photoprogram2/
current_font.jpg" , fn );
system( cmd12 );

/*Exposure*/
sprintf( cmd13 , "/usr/bin/convert -font
helvetica -fill white -pointsize 40 -draw
'text 1802,2090 \ "Exposure : %s\"' /iraf/iraf
/local/skymon/photoprogram2/current_font.jpg
/iraf/iraf/local/skymon/photoprogram2/
current_font.jpg" , sp );
system( cmd13 );

/*ISO*/
sprintf( cmd14 , "/usr/bin/convert -font
helvetica -fill white -pointsize 40 -draw
'text 1715,2130 \ "ISO-sensitivity : %s\"'
/iraf/iraf/local/skymon/photoprogram2/
current_font.jpg /iraf/iraf/local/skymon
/photoprogram2/current_font.jpg" , iso );
system( cmd14 );

/*whitebalance*/
sprintf( cmd16 , "/usr/bin/convert -font
helvetica -fill white -pointsize 40 -draw
'text 1739,2170 \ "whitebalance : %s\"'
/iraf/iraf/local/skymon/photoprogram2/
current_font.jpg /iraf/iraf/local/skymon
/photoprogram2/current_completed.jpg" , wh );
system( cmd16 );

/*名前を付けてディレクトリへ保存*/
sprintf( cmd8 , "cp -p /iraf/iraf/local
/skymon/photoprogram2/current_completed.jpg
/iraf/iraf/local/skymon/%s/%s/%s/%s.jpg\n"
, name2 , name3 , name4 , name1 );
system( cmd8 );

/*圧縮する*/
sprintf( cmd9 , "/usr/bin/convert -geometry
600x600 /iraf/iraf/local/skymon/photoprogram2
/current_completed.jpg /iraf/iraf/local/skymon
/photoprogram2/current_srk.jpg" );
system( cmd9 );

/*scp*/
sprintf( cmd15 , "/iraf/iraf/local/scpwebserver.sh" );
system( cmd15 );

/*smokaへコピー*/
if( strcmp( name6 , "1130" ) == 0 ){
sprintf( cmd18 , "mkdir /iraf/iraf/local/skymon/sm/sm%s"
, name4 );
}

smok = atoi( name6 );
smo = smok%10;

if ( smo == 0 ){

timer -= ( 86400/2 );
t_st = localtime(&timer);
sprintf( name9 , "%04d%02d%02d" , t_st->tm_year+1900 ,
t_st->tm_mon+1 , t_st->tm_mday );

sprintf( cmd1 , "cp -p /iraf/iraf/local/skymon
/photoprogram2/current_completed.jpg /iraf/iraf/
local/skymon/sm/sm%s/%s.jpg" , name9 , name7 );
system( cmd1 );
}

}

return 0;

```

付録 A.2 露出時間計算プログラム

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>

char sp;

int shutterspeed( int a , int b , int x )
{
    double d,n,o;
    double c ,rad,e;
    double y;
    int p,q,u,r,sh,bef,aft;
    double k,l;
    char j[30];
    char cmdl[100];

    /* 設定 */
    /* 変更する時間(日の出 bef 分前から aft 分後) */
    bef = 50;
    aft = 25;

    /* 昼の露出時間(k/l になるように) */
    k = 1.0;
    l = 800.0;
    p = 7;
    /* 設定番号 */
    /* 夜の露出時間 */
    char m[30] = "15.0"; /* 時間 */
    q = 48; /* 設定番号 */

    /* 設定番号は次から選んで下さい */
    /*
    0:1/4000    18:1/60    36:1.0
    1:1/3200    19:1/50    37:1.3
    2:1/2500    20:1/40    38:1.6
    3:1/2000    21:1/30    39:2.0
    4:1/1600    22:1/25    40:2.5
    5:1/1250    23:1/20    41:3.0
    6:1/1000    24:1/15    42:4.0
    7:1/800     25:1/13    43:5.0
    8:1/640     26:1/10    44:6.0
    9:1/500     27:1/8     45:8.0
    10:1/400    28:1/6     46:10.0
    11:1/320    29:1/5     47:13.0
    12:1/250    30:1/4     48:15.0
    13:1/200    31:1/3     49:20.0
    14:1/160    32:1/2.5   50:25.0
    15:1/125    33:1/2     51:30.0
    16:1/100    34:1/1.6   52:429496.7295
    17:1/80     35:1/1.3
    */

    u = k;
    r = l;
    sprintf( j , "%d/%d" , u , r );

    if ( 0 <= x && x < -(bef+10)*60+a ){
        sh = q;
        sprintf( &sp , "%ss" , m );
        return sh;
    }

    else if ( -(bef+10)*60+a <= x && x <
    -bef*60+a ){
        sh = q;
        sprintf( &sp , "%ss" , m );
        sprintf( cmdl , "/usr/local/bin/gphoto2
    --quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( aft*60+a < x && x <= (aft+10)*60+a ){
        sh = p;
        sprintf( &sp , "%ss" , j );
        sprintf( cmdl , "/usr/local/bin/gphoto2
    --quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
    }
}

```

```

return sh;
}

else if ( (aft+10)*60+a < x && x <
-(aft+10)*60+b ){
    sh = p;
    sprintf( &sp , "%ss" , j );
    return sh;
}

else if ( -(aft+10)*60+b <= x && x < -aft*60+b ){
    sh = p;
    sprintf( &sp , "%ss" , j );
    sprintf( cmdl , "/usr/local/bin/gphoto2
    --quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( bef*60+b < x && x <= (bef+10)*60+b ){
    sh = q;
    sprintf( &sp , "%ss" , m );
    sprintf( cmdl , "/usr/local/bin/gphoto2
    --quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( (bef+10)*60+b < x && x <= 86400 ){
    sh = q;
    sprintf( &sp , "%ss" , m );
    return sh;
}

else if ( -bef*60+a <= x && x <= aft*60+a ){
    n = k/l;
    o = atof(m);
    c=(log10(o)-log10(n))/2;
    e=(log10(o)+log10(n))/2;
    d=(x-a)/60;
    rad=(d+(bef-aft)/2)*M_PI/(bef+aft);
    y=c*sin(rad)+e;

    if ( log10(30) <= y ){
        sh = 51;
        sprintf( &sp , "30s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
    --quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(25) <= y && y < log10(30) ){
        sh = 50;
        sprintf( &sp , "25s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
    --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(20) <= y && y < log10(25) ){
        sh = 49;
        sprintf( &sp , "20s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
    --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(15) <= y && y < log10(20) ){
        sh = 48;
        sprintf( &sp , "15s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
    --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }
}

```

```

}
else if ( log10(13) <= y && y < log10(15) ){
sh = 47;
sprintf( &sp , "13s" );
return sh;
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(10) <= y && y < log10(13) ){
sh = 46;
sprintf( &sp , "10s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(8) <= y && y < log10(10) ){
sh = 45;
sprintf( &sp , "8s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(6) <= y && y < log10(8) ){
sh = 44;
sprintf( &sp , "6s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(5) <= y && y < log10(6) ){
sh = 43;
sprintf( &sp , "5s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(4) <= y && y < log10(5) ){
sh = 42;
sprintf( &sp , "4s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(3) <= y && y < log10(4) ){
sh = 41;
sprintf( &sp , "3s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(2.5) <= y && y < log10(3) ){
sh = 40;
sprintf( &sp , "2.5s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(2) <= y && y < log10(2.5) ){
sh = 39;
sprintf( &sp , "2s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.6) <= y && y < log10(2) ){
sh = 38;
sprintf( &sp , "1.6s" );

```

```

sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.3) <= y && y < log10(1.6) ){
sh = 37;
sprintf( &sp , "1.3s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1) <= y && y < log10(1.3) ){
sh = 36;
sprintf( &sp , "1s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/1.3) <= y && y < log10(1) ){
sh = 35;
sprintf( &sp , "1/1.3s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/1.6) <= y && y <
log10(1.0/1.3) ){
sh = 34;
sprintf( &sp , "1/1.6s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/2.0) <= y && y <
log10(1.0/1.6) ){
sh = 33;
sprintf( &sp , "1/2s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/2.5) <= y && y <
log10(1.0/2.0) ){
sh = 32;
sprintf( &sp , "1/2.5s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/3.0) <= y && y <
log10(1.0/2.5) ){
sh = 31;
sprintf( &sp , "1/3s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/4.0) <= y && y <
log10(1.0/3.0) ){
sh = 30;
sprintf( &sp , "1/4s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmd1 );
return sh;
}

else if ( log10(1.0/5.0) <= y && y <
log10(1.0/4.0) ){
sh = 29;
sprintf( &sp , "1/5s" );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );

```

```

system( cmdl );
return sh;
}

else if ( log10(1.0/6.0) <= y && y <
log10(1.0/5.0) ){
sh = 28;
sprintf( &sp , "1/6s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/8.0) <= y && y <
log10(1.0/6.0) ){
sh = 27;
sprintf( &sp , "1/8s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/10.0) <= y && y <
log10(1.0/8.0) ){
sh = 26;
sprintf( &sp , "1/10s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/13.0) <= y && y <
log10(1.0/10.0) ){
sh = 25;
sprintf( &sp , "1/13s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/15.0) <= y && y <
log10(1.0/13.0) ){
sh = 24;
sprintf( &sp , "1/15s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/20.0) <= y && y <
log10(1.0/15.0) ){
sh = 23;
sprintf( &sp , "1/20s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/25.0) <= y && y <
log10(1.0/20.0) ){
sh = 22;
sprintf( &sp , "1/25s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/30.0) <= y && y <
log10(1.0/25.0) ){
sh = 21;
sprintf( &sp , "1/30s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/40.0) <= y && y <
log10(1.0/30.0) ){
sh = 20;
sprintf( &sp , "1/40s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/50.0) <= y && y <
log10(1.0/40.0) ){
sh = 19;
sprintf( &sp , "1/50s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/60.0) <= y && y <
log10(1.0/50.0) ){
sh = 18;
sprintf( &sp , "1/60s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/80.0) <= y && y <
log10(1.0/60.0) ){
sh = 17;
sprintf( &sp , "1/80s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/100.0) <= y && y <
log10(1.0/80.0) ){
sh = 16;
sprintf( &sp , "1/100s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/125.0) <= y && y <
log10(1.0/100.0) ){
sh = 15;
sprintf( &sp , "1/125s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/160.0) <= y && y <
log10(1.0/125.0) ){
sh = 14;
sprintf( &sp , "1/160s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/200.0) <= y && y <
log10(1.0/160.0) ){
sh = 13;
sprintf( &sp , "1/200s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/250.0) <= y && y <
log10(1.0/200.0) ){
sh = 12;
sprintf( &sp , "1/250s" );
printf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/320.0) <= y && y <
log10(1.0/250.0) ){
sh = 11;
sprintf( &sp , "1/320s" );

```

```

        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/400.0) <= y && y <
log10(1.0/320.0) ){
        sh = 10;
        sprintf( &sp , "1/400s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/500.0) <= y && y <
log10(1.0/400.0) ){
        sh = 9;
        sprintf( &sp , "1/500s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/640.0) <= y && y <
log10(1.0/500.0) ){
        sh = 8;
        sprintf( &sp , "1/640s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/800.0) <= y && y <
log10(1.0/640.0) ){
        sh = 7;
        sprintf( &sp , "1/800s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/1000.0) <= y && y <
log10(1.0/800.0) ){
        sh = 6;
        sprintf( &sp , "1/1600s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/1250.0) <= y && y <
log10(1.0/1000.0) ){
        sh = 5;
        sprintf( &sp , "1/1250s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/1600.0) <= y && y <
log10(1.0/1250.0) ){
        sh = 4;
        sprintf( &sp , "1/1600s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/2000.0) <= y && y <
log10(1.0/1600) ){
        sh = 3;
        sprintf( &sp , "1/2000s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/2500.0) <= y && y <
log10(1.0/2000.0) ){
        sh = 2;
        sprintf( &sp , "1/2500s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    else if ( log10(1.0/3200.0) <= y && y <
log10(1.0/2500.0) ){
        sh = 1;
        sprintf( &sp , "1/3200s" );
        sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmdl );
        return sh;
    }

    }

    else if ( -aft*60+b <= x && x <= bef*60+b ){
        n = k/1;
        o = atof(m);
        c=(log10(o)-log10(n))/2;
        e=(log10(o)+log10(n))/2;
        d=(x-b)/60;
        rad=(d+(aft-bef)/2)*M.PI/(bef+aft);
        y=c*sin(rad)+e;

        if ( log10(30) <= y ){
            sh = 51;
            sprintf( &sp , "30s" );
            sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
            system( cmdl );
            return sh;
        }

        else if ( log10(25) <= y && y < log10(30) ){
            sh = 50;
            sprintf( &sp , "25s" );
            sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
            system( cmdl );
            return sh;
        }

        else if ( log10(20) <= y && y < log10(25) ){
            sh = 49;
            sprintf( &sp , "20s" );
            sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
            system( cmdl );
            return sh;
        }

        else if ( log10(15) <= y && y < log10(20) ){
            sh = 48;
            sprintf( &sp , "15s" );
            sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
            system( cmdl );
            return sh;
        }

        else if ( log10(13) <= y && y < log10(15) ){
            sh = 47;
            sprintf( &sp , "13s" );
            return sh;
            sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , 47 );
            system( cmdl );
            return sh;
        }

        else if ( log10(10) <= y && y < log10(13) ){
            sh = 46;
            sprintf( &sp , "10s" );
            sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet

```

```

--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(8) <= y && y < log10(10) ){
sh = 45;
sprintf( &sp , "8s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(6) <= y && y < log10(8) ){
sh = 44;
sprintf( &sp , "6s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(5) <= y && y < log10(6) ){
sh = 43;
sprintf( &sp , "5s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(4) <= y && y < log10(5) ){
sh = 42;
sprintf( &sp , "4s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(3) <= y && y < log10(4) ){
sh = 41;
sprintf( &sp , "3s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(2.5) <= y && y < log10(3) ){
sh = 40;
sprintf( &sp , "2.5s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(2) <= y && y < log10(2.5) ){
sh = 39;
sprintf( &sp , "2s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.6) <= y && y < log10(2) ){
sh = 38;
sprintf( &sp , "1.6s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.3) <= y && y < log10(1.6) ){
sh = 37;
sprintf( &sp , "1.3s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1) <= y && y < log10(1.3) ){
sh = 36;
sprintf( &sp , "1s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/1.3) <= y && y < log10(1) ){
sh = 35;
sprintf( &sp , "1/1.3s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/1.6) <= y && y <
log10(1.0/1.3) ){
sh = 34;
sprintf( &sp , "1/1.6s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/2.0) <= y && y <
log10(1.0/1.6) ){
sh = 33;
sprintf( &sp , "1/2s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/2.5) <= y && y <
log10(1.0/2.0) ){
sh = 32;
sprintf( &sp , "1/2.5s" );
sprintf( cmdl , "/usr/local/bin/gphoto2 --quiet
--set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/3.0) <= y && y <
log10(1.0/2.5) ){
sh = 31;
sprintf( &sp , "1/3s" );
sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/4.0) <= y && y <
log10(1.0/3.0) ){
sh = 30;
sprintf( &sp , "1/4s" );
sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/5.0) <= y && y <
log10(1.0/4.0) ){
sh = 29;
sprintf( &sp , "1/5s" );
sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/6.0) <= y && y <
log10(1.0/5.0) ){
sh = 28;
sprintf( &sp , "1/6s" );
sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
system( cmdl );
return sh;
}

else if ( log10(1.0/8.0) <= y && y <
log10(1.0/6.0) ){

```

```

    sh = 27;
    sprintf( &sp , "1/8s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/10.0) <= y && y <
log10(1.0/8.0) ){
    sh = 26;
    sprintf( &sp , "1/10s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/13.0) <= y && y <
log10(1.0/10.0) ){
    sh = 25;
    sprintf( &sp , "1/13s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/15.0) <= y && y <
log10(1.0/13.0) ){
    sh = 24;
    sprintf( &sp , "1/15s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/20.0) <= y && y <
log10(1.0/15.0) ){
    sh = 23;
    sprintf( &sp , "1/20s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/25.0) <= y && y <
log10(1.0/20.0) ){
    sh = 22;
    sprintf( &sp , "1/25s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/30.0) <= y && y <
log10(1.0/25.0) ){
    sh = 21;
    sprintf( &sp , "1/30s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/40.0) <= y && y <
log10(1.0/30.0) ){
    sh = 20;
    sprintf( &sp , "1/40s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/50.0) <= y && y <
log10(1.0/40.0) ){
    sh = 19;
    sprintf( &sp , "1/50s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/60.0) <= y && y <
log10(1.0/50.0) ){
    sh = 18;
    sprintf( &sp , "1/60s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/80.0) <= y && y <
log10(1.0/60.0) ){
    sh = 17;
    sprintf( &sp , "1/80s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/100.0) <= y && y <
log10(1.0/80.0) ){
    sh = 16;
    sprintf( &sp , "1/100s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/125.0) <= y && y <
log10(1.0/100.0) ){
    sh = 15;
    sprintf( &sp , "1/125s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/160.0) <= y && y <
log10(1.0/125.0) ){
    sh = 14;
    sprintf( &sp , "1/160s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/200.0) <= y && y <
log10(1.0/160.0) ){
    sh = 13;
    sprintf( &sp , "1/200s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/250.0) <= y && y <
log10(1.0/200.0) ){
    sh = 12;
    sprintf( &sp , "1/250s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/320.0) <= y && y <
log10(1.0/250.0) ){
    sh = 11;
    sprintf( &sp , "1/320s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}

else if ( log10(1.0/400.0) <= y && y <
log10(1.0/320.0) ){
    sh = 10;
    sprintf( &sp , "1/400s" );
    sprintf( cmdl , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
    system( cmdl );
    return sh;
}
}

```

```

    else if ( log10(1.0/500.0) <= y && y <
log10(1.0/400.0) ){
        sh = 9;
        sprintf( &sp , "1/500s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/640.0) <= y && y <
log10(1.0/500.0) ){
        sh = 8;
        sprintf( &sp , "1/640s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/800.0) <= y && y <
log10(1.0/640.0) ){
        sh = 7;
        sprintf( &sp , "1/800s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/1000.0) <= y && y <
log10(1.0/800.0) ){
        sh = 6;
        sprintf( &sp , "1/1600s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/1250.0) <= y && y <
log10(1.0/1000.0) ){
        sh = 5;
        sprintf( &sp , "1/1250s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/1600.0) <= y && y <
log10(1.0/1250.0) ){
        sh = 4;
        sprintf( &sp , "1/1600s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/2000.0) <= y && y <
log10(1.0/1600.0) ){
        sh = 3;
        sprintf( &sp , "1/2000s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/2500.0) <= y && y <
log10(1.0/2000.0) ){
        sh = 2;
        sprintf( &sp , "1/2500s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

    else if ( log10(1.0/3200.0) <= y && y <
log10(1.0/2500.0) ){
        sh = 1;
        sprintf( &sp , "1/3200s" );
        printf( cmd1 , "/usr/local/bin/gphoto2
--quiet --set-config=shutterspeed=%d" , sh );
        system( cmd1 );
        return sh;
    }

```

```

    }
    return sh;
}

return 0;
}

付録 A.3 F 値計算プログラム

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
char fn[30];

int fnumber( int a , int b ,int x )
{
    int f;
    double d,n,o;
    double c,rad,e;
    double y;
    int p,q,bef,aft;
    char cmd2[100];

/* 設定 */
/* 昼の F 値 */
/* 変更する時間 (日の出 bef 分前から aft 分後) */
    bef = 50;
    aft = 60;

    char l[30] = "8"; /* F 値 (f は省略) */
    p = 9; /* 設定番号 */
    /* 夜の F 値 */
    char m[30] = "3.2"; /* F 値 (f は省略) */
    q = 1; /* 設定番号 */

/* 設定番号は次から選んで下さい */
/* (0:f/2.8) 7:f/6.3 14:f/14
1:f/3.2 (8:f/7.1) 15:f/16
2:f/3.5 9:f/8 16:f/18
3:f/4 10:f/9 17:f/20
4:f/4.5 11:f/10 18:f/22
5:f/5 12:f/11
(6:f/5.6) 13:f/13 */

    if ( 0 <= x && x < -(bef+10)*60+a ){
        f = q;
        sprintf( fn , "f/%s" , m );
        return f;
    }

    else if ( -(bef+10)*60+a <= x && x <= -bef*60+a ){
        f = q;
        sprintf( fn , "f/%s" , m );
        sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
        system( cmd2 );
        return f;
    }

    else if ( aft*60+a <= x && x < (aft+10)*60+a ){
        f = p;
        sprintf( fn , "f/%s" , l );
        sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
        system( cmd2 );
        return f;
    }

    else if ( (aft+10)*60+a < x && x < -(aft+10)*60+b ){
        f = p;
        sprintf( fn , "f/%s" , l );

```

```

return f;
}

else if ( -(aft+10)*60+b <= x && x <= -aft*60+b ){
f = p;
sprintf( fn , "f/%s" , l );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( bef*60+b <= x && x <= (bef+5)*60+b ){
f = q;
sprintf( fn , "f/%s" , m );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( (bef+5)*60+b < x && x <= 86400 ){
f = q;
sprintf( fn , "f/%s" , m );
return f;
}

else if ( -bef*60+a <= x && x <= aft*60+a ){
n = atof(l);
o = atof(m);

c=(18/o-18/n)/2;
e=(18/n+18/o)/2;
d=(x-a)/60;
rad=(d+(bef-aft)/2)*M.PI/(bef+aft);
y=-c*sin(rad)+e;

if ( 18/2.8 <= y ){
f = 0;
sprintf( fn , "f/2.8" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/3.2 <= y && y < 18/2.8 ){
f = 1;
sprintf( fn , "f/3.2" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/3.5 <= y && y < 18/3.2 ){
f = 2;
sprintf( fn , "f/3.5" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/4 <= y && y < 18/3.5 ){
f = 3;
sprintf( fn , "f/4" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/4.5 <= y && y < 18/4 ){
f = 4;
sprintf( fn , "f/4.5" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

}

else if ( 18/5 <= y && y < 18/4.5 ){
f = 5;
sprintf( fn , "f/5" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

/* else if ( 18/5.6 <= y && y < 18/5 ){
f = 6;
sprintf( fn , "f/5.6" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

*/

else if ( 18/6.3 <= y && y < 18/5 ){
f = 7;
sprintf( fn , "f/6.3" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

/* else if ( 18/7.1 <= y && y < 18/6.3 ){
f = 8;
sprintf( fn , "f/7.1" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

*/

else if ( 18/8 <= y && y < 18/6.3 ){
f = 9;
sprintf( fn , "f/8" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/9 <= y && y < 18/8 ){
f = 10;
sprintf( fn , "f/9" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/10 <= y && y < 18/9 ){
f = 11;
sprintf( fn , "f/10" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/11 <= y && y < 18/10 ){
f = 12;
sprintf( fn , "f/11" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/13 <= y && y < 18/11 ){
f = 13;
sprintf( fn , "f/13" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/14 <= y && y < 18/13 ){
f = 14;
sprintf( fn , "f/14" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet

```

```

--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/16 <= y && y <18/14 ){
f = 15;
sprintf( fn , "f/16" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/18 <= y && y <18/16 ){
f = 16;
sprintf( fn , "f/18" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/20 <= y && y <18/18 ){
f = 17;
sprintf( fn , "f/20" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/22 <= y && y <18/20 ){
f = 18;
sprintf( fn , "f/22" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

return 0;
}

else if ( -aft*60+b <= x && x <= bef*60+b ){

n = atof(l);
o = atof(m);

c=(18/o-18/n)/2;
e=(18/n+18/o)/2;
d=(x-b)/60;
rad=(d+(aft-bef)/2)*M.PI/(bef+aft);
y=c*sin(rad)+e;

if ( 18/2.8 <= y ){
f = 0;
sprintf( fn , "f/2.8" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/3.2 <= y && y < 18/2.8 ){
f = 1;
sprintf( fn , "f/3.2" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/3.5 <= y && y <18/3.2 ){
f = 2;
sprintf( fn , "f/3.5" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/4 <= y && y <18/3.5 ){
f = 3;
sprintf( fn , "f/4" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/4.5 <= y && y <18/4 ){
f = 4;
sprintf( fn , "f/4.5" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/5 <= y && y <18/4.5 ){
f = 5;
sprintf( fn , "f/5" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

/* else if ( 18/5.6 <= y && y <18/5 ){
f = 6;
sprintf( fn , "f/5.6" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}*/

else if ( 18/6.3 <= y && y <18/5 ){
f = 7;
sprintf( fn , "f/6.3" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

/* else if ( 18/7.1 <= y && y <18/6.3 ){
f = 8;
sprintf( fn , "f/7.1" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
return f;
}*/

else if ( 18/8 <= y && y <18/6.3 ){
f = 9;
sprintf( fn , "f/8" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/9 <= y && y <18/8 ){
f = 10;
sprintf( fn , "f/9" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/10 <= y && y <18/9 ){
f = 11;
sprintf( fn , "f/10" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/11 <= y && y <18/10 ){
f = 12;
sprintf( fn , "f/11" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet

```

```

--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/13<= y && y <18/11 ){
f = 13;
sprintf( fn , "f/13" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/14 <= y && y <18/13 ){
f = 14;
sprintf( fn , "f/14" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/16 <= y && y <18/14 ){
f = 15;
sprintf( fn , "f/16" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/18 <= y && y <18/16 ){
f = 16;
sprintf( fn , "f/18" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/20 <= y && y <18/18 ){
f = 17;
sprintf( fn , "f/20" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

else if ( 18/22 <= y && y <18/20 ){
f = 18;
sprintf( fn , "f/22" );
sprintf( cmd2 , "/usr/local/bin/gphoto2 --quiet
--set-config=f-number=%d" , f );
system( cmd2 );
return f;
}

return 0;
}

return 0;
}

```

付録 A.4 ISO 感度計算プログラム

```

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
char iso[30];

int sensitivity( int a , int b ,int x )
{
int is;
char cmd1[100];
double d,n,o;
double c,rad,e;
double y;
int p,q,u,r,l,m,bef,aft;

/* 設定*/
/*変更する時間(日の出 bef分前からaft分後)*/

```

```

bef = 50;
aft = 25;

/* 昼の iso*/
l = 200; /* iso*/
p = 1; /* 設定番号*/
/* 夜の iso*/
m = 1600; /* iso*/
q = 4; /* 設定番号*/

/* 設定番号は次から選んで下さい*/
/* 0:100
1:200
2:400
3:800
4:1600
5:3200 */

if ( 0 <= x && x < -(bef+10)*60+a ){
is = q;
sprintf( iso , "%d" , m );
return is;
}

else if ( -(bef+10)*60+a <= x && x < -bef*60+a ){
is = q;
sprintf( iso , "%d" , m );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
system( cmd1 );
return is;
}

else if ( aft*60+a < x && x <= (aft+10)*60+a ){
is = p;
sprintf( iso , "%d" , l );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
system( cmd1 );
return is;
}

else if ( (aft+10)*60+a < x && x < -(aft+10)*60+b ){
is = p;
sprintf( iso , "%d" , l );
return is;
}

else if ( -(aft+10)*60+b <= x && x < -aft*60+b ){
is = p;
sprintf( iso , "%d" , l );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
system( cmd1 );
return is;
}

else if ( bef*60+b <= x && x <= (bef+10)*60+b ){
is = q;
sprintf( iso , "%d" , m );
sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
system( cmd1 );
return is;
}

else if ( (bef+10)*60+b < x && x <= 86400 ){
is = q;
sprintf( iso , "%d" , m );
return is;
}

else if ( -bef*60+a <= x && x <= aft*60+a ){
c=(m-1)/2;
e=(l+m)/2;
d=(x-a)/60;
rad=(d+(bef-aft)/2)*M_PI/(bef+aft);
y=-c*sin(rad)+e;

```

```

if ( 3200 <= y ){
    is = 5;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "3200" );
    return is;
}

else if ( 1600 <= y && y < 3200 ){
    is = 4;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "1600" );
    return is;
}

else if ( 800 <= y && y < 1600 ){
    is = 3;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "800" );
    return is;
}

else if ( 400 <= y && y < 800 ){
    is = 2;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "400" );
    return is;
}

else if ( 200 <= y && y < 400 ){
    is = 1;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "200" );
    return is;
}

else if ( 100 <= y && y < 200 ){
    is = 0;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "100" );
    return is;
}

return 0;
}

else if ( -aft*60+b <= x && x <= bef*60+b ){
    c=(m-1)/2;
    e=(1+m)/2;
    d=(x-b)/60;
    rad=(d+(aft-bef)/2)*M_PI/(bef+aft);
    y=c*sin(rad)+e;
}

if ( 3200 <= y ){
    is = 5;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "3200" );
    return is;
}

else if ( 1600 <= y && y < 3200 ){
    is = 4;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "1600" );
    return is;
}

else if ( 800 <= y && y < 1600 ){
    is = 3;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "800" );
    return is;
}

else if ( 400 <= y && y < 800 ){
    is = 2;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "400" );
    return is;
}

else if ( 200 <= y && y < 400 ){
    is = 1;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "200" );
    return is;
}

else if ( 100 <= y && y < 200 ){
    is = 0;
    sprintf( cmd1 , "/usr/local/bin/gphoto2 --quiet
--set-config=iso=%d" , is );
    system( cmd1 );
    sprintf( iso , "100" );
    return is;
}

return 0;
}

return 0;
}
}

```

謝辞

本研究において、たくさんの方々にご協力して頂きました。特に指導教官の川端先生には、東広島天文台への移動、寒い中での装置の加工や設置、プログラム作成におけるアドバイスなどお忙しいスケジュールの中、親切なご指導をして頂き、本当にありがとうございます。前任者の奥嶋さん、カメラなどの道具の準備や、gphoto2の使用方法等を丁寧に教えて頂きました。ありがとうございます。また、自分の研究があるにもかかわらず、私のわからないことにわざわざ時間をさいて、丁寧に教えて下さった、研究室の皆さん本当にありがとうございます。

授業やセミナー、日々の生活などを通して、深沢先生、大杉先生、吉田先生、水野先生、植村先生、片桐先生、ひろたかさんにも、本当にたくさんのことを教えて頂きました。ありがとうございます。

日常生活においても、部屋なんて関係なしに、優しくして頂いた、大部屋、お茶部屋、大杉部屋の皆さん、また同じ部屋で楽しい時間を作ってくださった、小部屋の皆さん、一緒に遊んで頂いたり、わからないことやパソコンが壊れた時など、困った時にはいつも助けて頂いたり、本当にありがとうございました。



関連図書

- [1] 高橋 麻奈「やさしいC」風工舎 (2010)
- [2] 保田 知則「広島大学望遠鏡用全天スカイモニターと自動観測スケジュール機能の開発」2004 広島大学卒業論文
- [3] 上原 岳士「et-al,A&A,519,A56」(2010)
- [4] 浦口 史寛、吉田 道利、清水 康広、岡田 隆史「岡山天体物理観測所スカイモニターの開発」岡山天文台報 第4巻
- [5] cloudy Nights Telescope Reviews(http://www.cloudynights.com/item.php?item_id=1052)
- [6] ImageMagick(<http://mechanics.civil.tohoku.ac.jp/soft/node43.html>)