Studies of the Links Between Quasar Black Hole Accretion and Star Formation in Their Host Galaxies: Data Analysis of the Herschel Astrophysical Terahertz Large Area Survey

Thesis

UNRESTRICTED

Studies Of The Links Between Quasar Black Hole Accretion And Star Formation In Their Host Galaxies: Data Analysis Of The Herschel Astrophysical Terahertz Large Area Survey

by

M.D.D. Goodfellow

A thesis submitted

in partial fulfilment of the

requirements for the degree of

Master of *Philosophy*

Department of *Physical Sciences*

The Open University

April 2012

Milton Keynes, U.K.

DATE OF SUBMISSION: 10 MAY 2012

DATE OF AWARD: 11 JANUARY 2013

Abstract

The topic of this thesis is to gather evidence for a link between star formation in galaxies that contain a quasar in their nuclei. For quasars I argue that the best technique to measure the star formation rates is to use the far infrared.

I used the Herschel PACS 100 and 160 micron infrared data from the Herschel ATLAS programme to examine star formation rates in quasar host galaxies. I performed a stacking analysis of the PACS 100 and 160 micron data of a sample of 411 quasar candidates from the SDSS and 2SLAQ catalogues. The quasars are detected at up to 32σ, which I infer is from star formation in the galaxies that contain these quasars.

In order to obtain this data I have written a new timeline filtering subprogramme, plus a cross-scan filtering subprogramme, for the PACS data analysis pipeline. The outputs of this programme, using the early science phase Herschel data, were used to do the quasar stacking analysis.

I was also able to derive the formal minimum-variance point source filter using the calculus of variations with undertermined Langrangian multiplers. This kernel optimises the signal to noise but leads to false positive signals. I opted to use a modified kernel that whitens the power spectrum of the noise, and gives us a more reliable point source detection. However, the SNR penalty, for different PACS pixels and data sets, lies in the range 30% to 40%.

“ ”

— Department of Physical Sciences

Acknowledgements

I wish to thank my supervisor Doctor Stephen Serjeant for his patient and steadfast support over the last twenty six months. I have learnt alot from him in many ways. His asistance in my conducting of this research has been very helpful and he has helped me improve my knowledge of IDL.

My second supervisor Professor Glenn White has proved to be a valuable source of assistance. He has helped me through by offering a number of suggestions and observations.

My supervisor's post doctoral fellow Mattia Negrello has been a great source of ideas and advice. He also helped look after me when my supervisors where away.

Geoff Bradshaw, the computer manager of the department of physical sciences at the Open University, has been of great assistance with computer problems and related issues.

Roz Hopwood, my supervisor's former PhD student, now a post doctoral student at Imperial college London, has proved useful with IDL programming and PACS pipeline programme issues.

Last and by no means least, I wish to thank my younger brother Guy Goodfellow, for his help in writing my first year probationary report and general support.

...

“ ”

— Department of Physical Sciences

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Preview of chapter one

In chapter one the theoretical back ground necessary for understanding star formation in galaxies that contain Quasars in their nuclei. In this chapter the theoretical back ground of galaxy formation and evolution is given. The various types of star formation indicators are discussed. The observation of galaxy evolution is described. Models of AGN unification are outlined. The feedback phenomena is explained.

1.2 Galaxy evolution

1.2.1 Theoretical background

1.2.1.1 The theory of Jeans mass

Assume that we have a perfect sphere of radius R, uniform density ρ and mass M. Its total gravitational potential energy E_{GR} is,

$$E_{GR} = \frac{-3GM^2}{5R} \tag{1.1}$$

where G is the gravitational constant. We define E_K, to be the total kinetic energy of the gas molecules that make up the perfect sphere of gas. E_K is given by equation 1.2.

$$E_K = \frac{3NkT}{2} \tag{1.2}$$

Here, N is the total number of molecules that make up the perfect sphere of gas, k is Boltzmann's constant, T is the average absolute temperature of the perfect sphere of gas. We have assumed here that the molecules of gas are hydrogen atoms, with mass m_p, where m_p is the mass of a proton. According to the virial theorem, the condition for gravitational collapse is given equation 1.3.

$$2E_K < -E_{GR} \tag{1.3}$$

This leads to the mass of the cloud being greater than the Jeans mass M_J, for gravitational collapse to occur. M_J is given by equation 1.4.

$$M_J = \left(\frac{5kT}{Gm_p}\right)^{3/2} \left(\frac{3}{4\pi\rho}\right)^{1/2} \tag{1.4}$$

This leads to the condition for gravitational collapse being given by equation 1.5.

$$M > M_J \tag{1.5}$$

1.2.1.2 Hierarchical structure formation of dark matter

The universe is mainly composed of dark matter which does not absorb or emit light. A piece of good observational evidence for dark matter is the rotational curves of local galaxies. If we assume that the matter distribution in a given galaxy is spherically symmetric and the orbits of all the stars in that galaxy are perfect circles with their centres located at the galaxy centre, then the mass lying inside the orbit of a given star of radius r, *M(r)* is given by equation 1.6

$$M(r) = \frac{rv^2}{G}, \tag{1.6}$$

where v is the speed of the star moving in a perfectly circular orbit around the galactic centre of that galaxy and G is the gravitational constant. But, galaxies are not spherically symmetric, but we can still use the virial theorem mentioned in section 1.1.1.1. By producing a rotation curve for a local galaxy the shape of *M(r)* gives strong evidence for the presence of much more matter inside galaxies than can be explained by stars, gas and dust. The gas can be observed by fine structure radio transitions, for example HI. The dust can be seen by using submillimeter emission which can be seen by Herschel, Submillimeter Common User Bolometric Array, SCUBA, Atacama Large Millimeter/submillimeter Array and et cetera.

Currently dark matter can only be detected by its gravitational interactions.

The two best candidates for dark matter currently are the neutralino and the axion, that is non-baryonic matter, see Stephen Serjeant's book observational cosmology, Cambridge 2010, for details. The neutralino is the super symmetric partner of the neutrino. Super symmetry is an extension of the standard model of particle physics, which predicts that every particle should have a super symmetric partner.

The neutralino is an example of a weakly interacting massive particle, WIMP. If neutralinos exist in cosmologically significant numbers, they may be detected by observing the recoils of nuclei that have been hit by them. There are a number of experimental groups looking for WIMPs. The interaction probability of WIMPs with normal matter is predicted to be very tiny. But, with a high flux of WIMPs passing through the Earth, due to the motion of the Solar System through our Galaxy, there is a good chance that these rare collisions could be seen. Currently no WIMPs have been definitely detected by these experiments, see Stephen Serjeant's book observational cosmology, Cambridge 2010, for details. Axions are particles predicted by quantum chromodynamics, QCD. These particles have a very small mass, the mass is thought to be in the range 10^{-3} to 10^{-4} eV. Quantum chromodynamics is the gauge theory of the strong nuclear force that binds nuclei together and describes the strong interactions of hadrons, that is baryons or mesons, see Stephen Serjeant's book observational cosmology, Cambridge 2010, for details. Axions are thought to interact totally electromagnetically with a very tiny cross-section. Axions are thought to decay into a photon pair when they are in a strong magnetic field, see Stephen Serjeant's book observational cosmology, Cambridge 2010, for details. Currently there are no confirmed axion detections using this approach.

Tiny virialized dark matter haloes merge to generate larger dark matter haloes. This is known as bottom-up structure formation, as opposed to top-down structure formation that is due to the removal of small-scale structure that is due to HDM, Hot Dark Matter. It is vital to say that what occurs with baryons that comprise galaxies, stars, gas and dust, is totally different that which occurs for dark matter. In the early Universe, the time evolution of matter perturbations was effected by photon-baryon interactions, due to Thomson scattering.

1.2.1.3 Theoretical models of galaxy evolution

It is believed that galaxies formed from very small density fluctuations in the expanding cosmic gas produced by the Big Bang. These fluctuations grew in size as the areas of higher density stronger gravitational attractive force accumulated further matter. Spiral galaxies formed as a consequence of the gas cooling down and entering the areas of higher density, the discs of these spiral galaxies formed as a consequence of the angular momentum of this material. Elliptical galaxies formed as a result of mergers of smaller spiral galaxies or alternatively as a result of the gas cooling down in regions of low angular momentum. See Andrew Nortan's book, introduction to astrophysics, OU press 2002, for details.

Applying the deep exposures of apparently empty regions of the sky, for example the Hubble Deep Fields, very dim galaxies can now be observed, these galaxies are very distant as a result of being observed just after they formed in the young Universe. We can see, images of sub-galactic objects which could be in the process of merging to produce bigger structures which we now see in the local Universe. There is a large amount of evidence that galaxy interactions, such as mergers and collisions, are vital in generating the observed galaxy shapes. It is very probable that these processes produce the large number of distorted atypical galaxies that are observed. They also could be of more importance in generating the large number of elliptical galaxies in many galaxy clusters. See Andrew Nortan's book, introduction to astrophysics, OU press 2002, for details.

The accepted view of galaxy formation is that they form inside dark matter halos.

The halos form principally due to gravity and this process can be understound by using high-resolution cosmological numerical simulations plus analytic models. How the stellar structure of galaxies formed is more difficult to understand and the link between the galaxies and their dark matter halos is not clear. At present we do not have a detailed understanding of the process of galaxy formation and evolution. See Zehavi, Pattri and Zheng 2011 paper for details.

A useful technique to study galaxy formation that lie inside halos of dark matter is the Semi-Analytic Modeling (SAM) of galaxy formation, as given in Cole et al. 1994, 2000, Benson et al. 2003, Croton et al. 2006b and Bower et al. 2006. All these models, use N-body simulations and analytic models of baryonic evolution, to explain that dark matter should be populated with galaxies. In the SAM approach, galaxies change as their first stars evolve and new stars are born, Zehavi et al. 2011. When galaxies merge their stellar content changes. As a result of galaxies merging the galaxies involved form a new galaxy of larger mass than any of the galaxies present before the merger. Mass is conserved in the merger to good approximation. These models, include a number of feed back or pre-heating mechanisms, for example those due to star formation, active galactic nuclei or ionization induced by background ultraviolet light, also influence the galaxy evolution at different stages. All these models produce several observable properties of galaxies such as the galaxy stellar mass functions and luminosity functions, Croton et al. 2006, Bower et al. 2006, and Fontanot et al. 2009.

1.2.2 Star formation indicators

1.2.2.1 Lyman alpha emission line

The Lyman alpha emission line Lyα, is produced by the electron transition between the 2^2P, and 1^2S, orbitals of neutral hydrogen, HI. This transition results in a single photon being emitted having an energy of, 10.2eV, that has a wavelength,λ, of 1216$\mathring{A}$. In the atmospheres of short-lived hot massive stars their hot atmospheres produce an intense radiation field that have a large number of photons with energy larger than 13.6eV, $\lambda < 912\mathring{A}$, these photons will ionize the neutral hydrogen in the atmosphere. After a long enough time, the separate protons and electrons recombine generating continuous emission due to the recombination into the higher excited states of the hydrogen atom. Then a sequence of electron transitions occurs producing a series of discrete emission lines, the last of them produce Lyman series photons. This leads to, the emission line spectrum due to hydrogen being often observed superimposed on the spectra of young stellar populations and is used as a star formation indicator.

The star formation rate (SFR) is defined as the rate at which gas is changed into stars and, summed over a given single body, is usually measured in solar masses per year $M_{\odot}/yr$. In cosmology, the fundamental star formation quantity is the cosmic star formation rate per unit volume, or the star formation rate density ρ_{SFR} with units of $M_{\odot}/yrMpc^3$. This quantity is measured for a number of distinct times over the age of the universe, and gives the cosmic star formation history. This quantity is fundamentally connected with the forces controlling galaxy formation and evolution. Typically ρ_{SFR} is deduced by summing the SFR of all the galaxies in a specific volume of space.

The SFR deduced from the Lyα line is deduced from the Hα star formation indicator that has a more precise calibration, given by SFR(Hα)=7.9$\times10^{-42}L_{H\alpha}M_{\bigodot}/yr$ in this case $L_{H\alpha}$ is the luminosity of the Hα line in erg/s; for example, Kennicutt 1998. When case-B recombination occurs, the relevant Lyα luminosity of the source is given by $L_{Ly\alpha} = 8.7L_{H\alpha}$ and as a result SFR(Lyα)=9.1$\times10^{-43}L_{Ly\alpha}M_{\bigodot}$/yr.

1.2.2.2 Ultraviolet continuum emission

The SFRs, deduced from the optical and near-ultraviolet continuum, are relatively inaccurate and suffer from systematic errors due to reddening or from an inaccurate Initial Mass Function, IMF, the stellar population age, or metallicity effects in the star formation rate, Larson and Tinsley 1978. But, this approach can be used to test the mean SFR properties of a big galaxy sample in the case where high accuracy is not needed. This method should not be used in the case where the dust distribution, population number or the IMFs are likely to change appreciably in a sample.

The limitations of the approach described above can be eliminated if observational data is obtained in wavelength ranges where the whole spectrum is mainly from young stars this leads to the SFR being directly proportional to brightness. The ideal wavelength is, 1250-2500$\AA$, which is longer than the Lyα forest but is far enough away to minimize spectral interference from older stellar populations. This wavelength range can not be observed from the earth's surface for nearby galaxies $z < 0.5$, but this zone can be seen in the spectra galaxies with redshifts in the range $z \sim 1-5$.

The link between the UV brughtness over a given wavelength range and the SFR can be deduced from a number of models. The calibration techniques have been outlined by Buat et al. 1989, Deharveng et al. 1994, Leitherer et al. 1995b, Meurer et al. 1995, Cowie et al. 1997, and Madau et al. 1998, for the wavelength range 1500-2800$\AA$. All these calibration factors lie in the range of approximately 0.3 dex. When changes have been made so that the same reference wavelength and IMF are used, the bigest observed change was due to using different stellar libraries or different assumptions made in producing the star formation rates. It is necessary to get the total measurements of the SFRs of galaxies, it is usual to assume that the SFR does not change significantly for time scales much longer than the lifetimes of the main stars emitting UV. This time scale is less than one hundred million years. This assumption is known as the continuous star formation approximation. Changing the calibration factor of the Madau et al. 1998 to Salpeter's 1955 IMF with masses inside the range $0.1 M_{\odot}$ to $100 M_{\odot}$, produces the equation

$$SFR(M_{\odot}/y) = 1.4 \times 10^{-28} L_{\nu}(erg/sHz). \tag{1.7}$$

1.2.2.3 H alpha line emission

The usual approach to producing the calibration factor for converting the ionizing flux to the SFR is to use a model of evolutionary synthesis. Only stars with masses larger than ten solar masses and lifetimes less than twenty million years produce the vast majority of the total ionizing flux, hence the emission lines produce a virtually immediate indicator of the SFR, that is independent of prior star formation behavior. There are many calibrations that are currently used by several authors, Kennicutt 1983a, Gallagher et al. 1984, Kennicutt et al. 1994, Leitherer and Heckman 1995, and Madau et al. 1998. Making use of the stellar number counts and using the Salpeter IMF, for stars in the mass range $0.1 M_{\odot}$ to $100 M_{\odot}$, as was the case for deriving equation 1.7, and using the calibration factors of Kennicutt et al. 1994 and Madau et al. 1998, they produce

$$SFR(M_{\odot}/year) = 7.9 \times 10^{-42} L(H\alpha)(ergs/s) \tag{1.8}$$

$$SFR(M_{\odot}/year) = 1.08 \times 10^{-53} Q(H^0)(1/s) \tag{1.9}$$

where $Q(H^0)$ is the ionizing photon luminosity and the Hα calibration factor is evaluated for case B recombination at $T_e = 10000K$. The relevant conversion factor $L(Br\gamma)$, 8.2×10^{-40}, is the same as the factor relevant to Hα. Equation 1.8 produces SFRs that are 7% less than the usualy used calibration factor, Kennicutt 1983a, with a distinct calibration factor due to a mixture of more recent models of stars and IMF with a small change, Kennicutt et al. 1994. As a result of these different approaches, there is a notable range of the calibration factors that have been published, by about 30%. The biggest part of the change is due to different models of stellar evolution and their atmospheres.

1.2.2.4 Far infrared continuum emission

The usefulness of the far infrared brightness as star formation rate tracker depends on how new stars heat the dust and the optical depth of dust in star forming regions. The simplest practical situation is where new stars are the main generators of optical to ultraviolet emission of the spectrum and the dust has high opacity everywhere. In this case the far infrared brightness gives an accurate estimate of the total brightness of the starburst. In this extreme case the far infrared brightness is the best star formation rate tracker. In the dense circumnuclear starbursts, these conditions approximately apply.

The uncertainties discussed above affect the calibration of star formation rates using far infrared brightness, and there is likely that there is no one calibration that works for all galaxy types. But, the far infrared emission should give an accurate measure of the star formation rate in starburst galaxies that have dust surrounding their nuclei. The star formation rate verses far-IR luminosity L_{FIR}, connection is deduced by using synthesis models. In the optically thick case, all we need to do is simulate the total brightness of populations of stars. The largest uncertainty in the choice of the appropriate age of the stellar population; this may be obtained from the starburst time scale or by the time scale for dust dispersal, hence the $\tau \gg 1$ approximation no longer holds. There are a number of different calibrations that have been suggested by various authors using different assumptions for the star formation time scale, Hunter et al. 1986, Lehnert and Heckman 1996, Meurer et al. 1997, and Kennicutt 1998. Using the Leitherer and Heckman 1995 model, for continuous bursts with an age range of ten million to one hundred million years and using the IMF in the Kennicutt 1998 paper, leads to the relation:

$$SFR(M_{\odot}/year) = 4.5 \times 10^{-44} L_{FIR}(ergs/s)(starbursts) \tag{1.10}$$

where L_{FIR} refers to the infrared brightness summed over the full mid and far infrared spectrum, 8μm-1000μm. For starbursts the majority of the emission lies in the range 10μm to 120μm. It is important to state that different papers have used different definitions of L_{FIR}. Most of the calibrations that have been published lie in the range plus or minus 30% of equation 1.10. To be accurate, the relation given in equation 1.10 only works for starbursts which are younger than one hundred million years; this is where the applied approximations hold.

1.2.2.5 X-ray continuum emission

Applying the ASCA and Chandra measurements of close starburst galaxies and RXTE/ASM, ASCA and MIR-KVANT/TTM studies of the high-mass X-ray binary (HMXB) populations in the Milky Way and the large and small Magellanic Clouds, Grimm, Gilfanov and Sunyaev 2003, show that the number and total X-ray luminosity of HMXBs can be used to measure of the SFR of a galaxy. In Grimm, Gilfanov and Sunyaev 2003, they give the case that to within observational error of the data available at the time, a linear connection between HMXB number and star formation rate holds. The connection between the SFR and total luminosity of HMXBs has a non-linear relationship in the low-SFR region $L_X \propto SFR^{\sim 1.7}$, but becomes a linear connection for a suitably high star formation rate $SFR \gtrsim 4.5 M_{\odot}/yr$ for $M_* > 8M_{\odot}$. The non-linear $L_X - SFR$ dependence in the low-SFR limit is not linked to non-linear dependent phenomena in the populations of HMXB sources. It is due to the situation that is described in Grimm, Gilfanov and Sunyaev 2003 where they measure the total luminosity of a population of distinct sources which may be dominated by the small sample of the brightest sources. But, while more complex SFR-dependent effects may exist, over the whole range of SFRs the data is overall consistent with a general luminosity function of HMXBs that can be approximately fitted by a power law with a differential gradient of about 1.6, a cut-off X-ray luminosity at $L_X \sim few \times 10^{40} erg/s$, and a normalization proportional to the star formation rate.

Two cases were considered: case one is the link between the total brightness of HMXBs in a galaxy and the SFR in that galaxy, and case two is the link between the number of HMXBs in a given galaxy and the SFR in that same galaxy. In both cases there is a strong link to the star formation rate and these X-ray observations may be used as distinct star formation rate indicators.

The total number of high mass X-ray binaries is directly proportional to the star formation rate:

$$SFR(M_{\odot}/yr) = \frac{N(L > 2 \times 10^{38} erg/s)}{2.9}. \tag{1.11}$$

At big enough star formation rates $SFR \gtrsim 4.5M_{\odot}/yr$, $L_{2-10kev} \gtrsim 3 \times 10^{40}$ erg/s respectively, the X-ray brightness of a given galaxy from high mass X-ray binaries is directly proportional to the star formation rate:

$$SFR(M_{\odot}/yr) = \frac{L_{2-10kev}}{6.7 \times 10^{39}\ erg/s}. \tag{1.12}$$

At small enough star formation rates $SFR \lesssim 4.5M_{\odot}/yr$ $L_{2-10kev} \lesssim 3 \times 10^{40}$ erg/s, the L_X-SFR relation becomes non-linear:

$$SFR(M_{\odot}/yr) = \left(\frac{L_{2-10kev}}{2.6 \times 10^{39} erg/s}\right)^{0.6}. \tag{1.13}$$

1.2.2.6 Radio continuum emission

The lifetimes of the high mass stars that generate the majority of the radio emission from normal galaxies are much less than the Hubble time, hence the current radio luminosity is directly proportional to the recent star-formation rate. Biermann 1976 was the first to use thermal and non-thermal radio brightness observations in descriptive galaxy models since at the time the measurements could be made with similar or better accuracy than could be done in the optical. This new approach contrasted (B-V) colour with blue brightness to radio brightness ratios. Biermann made four important assumptions. One, the initial mass function is independent of time. Two, the initial mass function is a power law. Three, the precise form of the initial mass function is $\psi(M) \propto M^{-\gamma}$ with gradient $\gamma = 2.35$. Four, the maximum mass limit is M_U. Condon 1992 assumed that all stars that have a mass $M \geq M_{SN} \sim 6.7 M_{\odot}$, produce supernova remnants that emit radio energy, hence the radio supernova production rate is given by,

$$\nu_{SN} = \int_{M_{SN}}^{M_U} \psi(M) dM. \tag{1.14}$$

A simple derivation of the link between the supernova rate and the non-thermal radio brightness is deduced from the $\Sigma - D$ connection, Ulvestad in 1982. The data for a frequency of 408MHz in units of $\Sigma W/m^2\ Hz\ sr$ is roughly $10^{-15}[D(pc)]^{-3}$ and $D(pc) \sim 0.43 E_{50}^{1/5}[n(1/cm^3)]^{-1/5}[t(yr)]^{2/5}$, where E_{50} is the energy produced by the explosion in multiples of 10^{50}ergs, n is the particle density of the region near the supernova, and t is the time elapsed from the supernova explosion (see Clark and Caswell 1976 for details). If a supernova remnant is a radio source existing in its own adiabatic timescale, $\tau \sim 2 \times 10^4 E_{50}^{4/17}[n(1/cm^3)]^{-9/17}$, then

$$\left(\frac{L_N}{10^{22} W/Hz}\right) \sim 1.4 E_{50}^{-1/17} \left(\frac{n}{1/cm^3}\right)^{-2/17} \left(\frac{\nu_{SN}}{1/yr}\right) \tag{1.15}$$

at 408MHz.

Condon and Yin 1990, replaced equation 1.15, with the observed Galactic link between L_N and ν_{SN} which is,

$$\left(\frac{L_N}{10^{22} W/Hz}\right) \sim 13 \left(\frac{\nu}{GHz}\right)^{-\alpha} \left(\frac{\nu_{SN}}{1/yr}\right). \tag{1.16}$$

Equation 1.16 is most likely true for most normal galaxies. This is the case since any significant change in the ratio L_N/ν_{SN} from one galaxy to another would violate the observed far infrared to radio connection, Condon 1992. Plus Völk et al.

1989, have suggested that the energy production appearing as cosmic-rays from each supernova is the same as that being produced in the galaxy M82 which is an example of a starburst galaxy and the same sort of thing is occurring in our own Galaxy.

The total radio thermal and non-thermal brightnesses and, the FIR-to-radio ratio for the majority of normal galaxies can be understood by a simple model which needs only one externally specified parameter, the mean star formation rate for stars with mass larger than $5M_{\odot}$:

$$\left(\frac{SFR(M \geq 5M_{\odot})}{M_{\odot}/yr}\right) \equiv \int_{5M_{\odot}}^{M_U} M\psi(M)dM. \tag{1.17}$$

The expanded initial mass function given in Miller-Scalo 1979, $\psi(M) \propto M^{-5/2}$ is truncated at $M_U \sim 100M_{\odot}$. All stars with masses larger than $M_{SN} = 8M_{\odot}$, will eventually become radio supernovae, hence the radio supernovae rate ν_{SN} is strongly linked to the star formation rate:

$$\left(\frac{\nu_{SN}}{1/yr}\right) \sim 0.041 \left(\frac{SFR(M \geq 5M_{\odot})}{M_{\odot}/yr}\right). \tag{1.18}$$

The non-thermal brightness L_N can be deduced from equation 1.16; it is

$$\left(\frac{L_N}{W/Hz}\right) \sim 5.3 \times 10^{21} \left(\frac{\nu}{GHz}\right)^{-\alpha} \left(\frac{SFR(M \geq 5M_{\odot})}{M_{\odot}/yr}\right), \tag{1.19}$$

where $\alpha \sim 0.8$ is known as the non-thermal spectral index.

An estimate of the fraction of the radio energy emitted from thermal sources can be deduced from stellar models using the electron temperature of $T_e \sim 10^4 K$, and the Lyman continuum absorption can be neglected. Using these results from Kennicutt 1983a the following equation can be deduced.

$$\left(\frac{L_T}{W/Hz}\right) \sim 5.5 \times 10^{20} \left(\frac{\nu}{GHz}\right)^{-0.1} \left(\frac{SFR(M \geq 5M_{\odot})}{M_{\odot}/yr}\right). \tag{1.20}$$

Where L_T is the thermal brightness of a galaxy. The fraction of the thermal energy emitted deduced from equation 1.19 being divided by equation 1.20 is consistent with the observed mean value given in Kennicutt 1983a. The ionization rate is given by

$$\left(\frac{N_{UV}}{1/s}\right) \sim 3.5 \times 10^{53} \left(\frac{SFR(M \geq 5M_{\odot})}{M_{\odot}/yr}\right). \tag{1.21}$$

1.2.3 Observations of galaxy evolution

1.2.3.1 The blue cloud and the red sequence

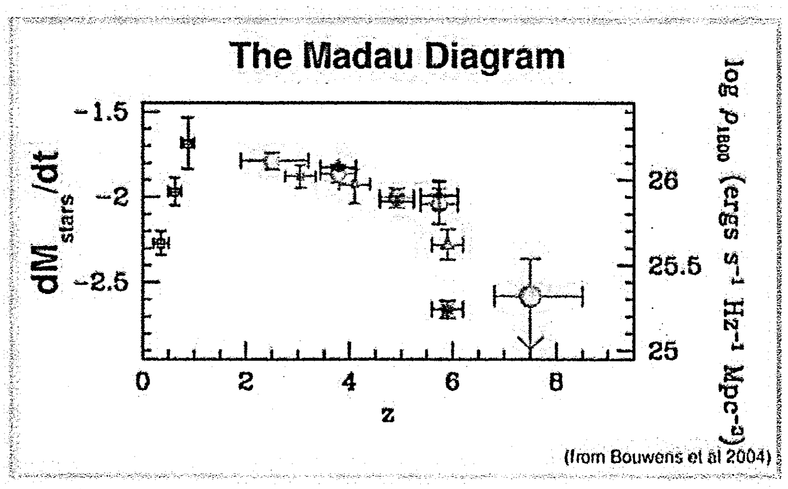
Stellar evolution is entirely specified by their movements on the Hertzsprung-Russell diagram, which is also known as the colour-magnitude diagram. A star's position on the Hertzsprung-Russell diagram can give a large amount of information about its chemical composition and internal structure, from birth to death. One of the key parts of a galaxy are the stars that lie inside them. In 2004 Eric Bell was the first person to construct colour-magnitude diagrams for galaxies. He also used them to make predictions about galaxy evolution. It should be made clear that a lot less information is given by the colour-magnitude diagrams for galaxies than in the stellar case. This is due to the fact that galaxy properties are not strongly determined by the mean luminosity and colour of their stars.

The arrangement of the positions of galaxies in their colour-magnitude diagram is useually divided into two regions, the blue cloud and the red sequence. Galaxies that lie in the blue cloud are more active than those in the red sequence, they are also undergoing star formation and they are of late Hubble type. Galaxies that lie in the red sequence are only passively evolving and they are of early Hubble type. This is true definitely for the local Universe. See Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details.

The sparsely populated narrow area between these two areas is known as the green valley. Some astronomers have used the red sequence as a general shape independent definition of early Hubble type galaxies, for galaxies that have redshifts less than one. The Galaxy Zoo project, Bamford et al. 2009, has given evidence for, in the local universe, that the colour-density link is stronger the than shape-density link. Evolution occurs in both the blue cloud and the red sequence, see Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details. The blue cloud is getting slightly redder with time, this may be due to stellar ageing or the increase of dust in the galaxies that lie in the blue cloud. There is a much larger number of bright blue cloud galaxies with redshifts greater than 0.5, than exist in the local universe. Similarly, the red sequence is getting redder with time, which is consistent with the expected normal stellar evolution. See Bamford et al. 2009 and McIntosh et al. 2005.

These discoveries may be due to the possibility that some of the galaxies that lie in the blue cloud that were forming stars stop star formation, and hence move to the red sequence and after that evolve passively. So the evolution of galaxies would follow the general pattern of forming in the blue cloud, (or forming outside the blue cloud) and as a result of galaxy mergers, starbursts begin so moving the galaxy into the blue cloud, and lastly moving through the green valley to meet the red sequence. But, this does not give us a mechanism for stopping star formation. It has been suggested that galaxies which lie in the green valley are active galaxies, see Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details, and this may give a means for ceassing star formation and the transitory nature of the green valley objects.

The Madau diagram, also known as the cosmic star formation history, is a graph



(from Bouwens et al 2004)

Figure 1.1: This is a Madau diagram or cosmic star formation history diagram, it is a plot of the star formation rate in solar masses per year per cubic megaparsec against redshift z.

of the star formation rate in solar masses per year per cubic megaparsec against redshift z. The star formation history is deduced from the luminosity of galaxies in the ultraviolet in their rest frames. A correction for dust absorption is applied . The Madau plot or diagram was first produced by Madau et al. in 1996. Madau was the first to give evidence of a high-redshift reduction in star formation rate. This diagram is sometimes called the Madau-Lilly diagram.

1.2.3.2 The luminosity function

The number of bodies per unit volume per unit luminosity L is called the luminosity function, usually given the symbol $\phi(L)$. In the case of high redshift galaxies a correction due to the expansion of the universe is applied. The usual approach is to replace the volume by the comoving volume, when deducing the luminosity functions. The form of luminosity functions is useually quite simple: a tiny gradient, which becomes steeper or exponential decay close to a special galaxy dependent luminosity L_* known as the break luminosity. The Schechter function is a good fit for many galaxy luminosity functions:

$$\phi(L) = \phi_* \left(\frac{L}{L_*}\right)^{-\alpha} exp\left(\frac{-L}{L_*}\right). \tag{1.22}$$

In the Schechter function above ϕ_* is the normalization parameter, α is the faint-end slope these parameters and the luminosity L_* are the parameters needed to specify a galaxy luminosity function. The Schechter function is very similar to shape of the Press-Schechter form of a galaxy luminosity function, see Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details; they are exactly the same if $\alpha = \frac{1}{2}$. This produces a white-noise spectrum and this is the main theoretical motivation for the Schechter function. But, the fundamental assumptions of the Press-Schechter form of the galaxy luminosity function are too simple to produce a perfect fit with the observed luminosity functions of galaxies. The overall shape of the Press-Schecter luminosity functions is a very close to the real galaxy luminosity functions.

1.2.3.3 Luminosity functions and the cosmic star formation history

The model described in the Baugh et al. 2004 paper explains how to evaluate the cosmic star formation history of galaxies. There exists a wide maximum, in the total star formation rate per unit volume for redshifts in the range two to three, which is the same as an emission time of roughly ten billion years. But, star formation from starbursts is about 5 per cent of the total star formation for all redshifts. The fraction of the formation of stars from starbursts rapidly gets larger with bigger redshift, and gets larger than normal star formation in discs for redshifts greater than about three, Le Floch et al. 2005. The higher starburst rate of star formation at large redshifts is due to the change in the star formation time-scale in quiescent discs, which produces an increase in gas richness at large redshifts. This also allows starbursts to be started in small mergers. But, the fraction of the formation of stars in total generated in the age of the universe is still small. Thirty per cent of the overall total of star formation is generated in starbursts according to the model given in Baugh et al. 2004. The mass contribution due to stars formed in star bursts is roughly 7% of the gross star mass today.

First the redshift z of a galaxy is obtained. Multiplying ϕ(L) with L gives the luminosity density, which next (at some wavelengths) is proportional to the SFR. Hence evaluating or estimating $\int_0^\infty L\phi(L,z)dL$ gives you the SFR. Thus we can calculate the star formation rate per unit volume taking place in a given galaxy. Since galaxies do not have a single locations on the Madau diagram a mean value is taken. Hence the galaxy's location on the Madau diagram is obtained. The Madau diagram is produced by plotting the log to the base ten of the star formation rate per unit volume, in solar masses per year per cubic megaparsec, for a sample of galaxies against their redshifts.

1.2.3.4 Downsizing

Up until the 1980s, many people thought there was a period in which all galaxies formed at some unknown redshift, even one huge collapse. The Hubble Space Telescope, HST, deep imaging changed the terminology used and the concepts used in describing galaxy formation, which lead to galaxy formation being thought of as an ongoing process, see Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details.

The discovery of Submillimetre Galaxies, SMGs with redshifts larger than two, with huge star formation rates, for instance thousands of solar masses per year, caused us to rethink galaxy formation models. The observational evidence became so strong to suggest that high mass galaxies formed before low mass galaxies. From hierarchical structure formation we would expect the opposite of this.

At the larger redshifts, massive starbursts form the largest part of the rate of cosmic star formation than is the case for smaller redshifts, Le Floch et al. 2005. This observation is surprising and is sometimes known as, 'anti-hierarchical'. But, semi-analytic models that are hierarchical can reproduce these observations, but they require unuseual IMFs or apply certain types of feedback mechanisms, see Stephen Serjeant's book, obervational cosmology, Cambridge 2010, for details. A frequently used and better term to describe the top-down means of galaxy formation is downsizing.

1.3 AGN unification

AGN, stands for Active Galactic Nuclei. In an AGN a large amount of energy is generated from non-stellar sources. There are two main types of AGN; quasars and radio galaxies. Radio galaxies produce electromagnetic radiation from a non-thermal source: synchrotron radiation produced by electrons circulating in a magnetic field, because they are accelerating due to the Lorentz force, $\overrightarrow{F} = q(\overrightarrow{v} \times \overrightarrow{B})$.

Quasar stands for quasi-stellar radio source, the first of which were discovered in 1960. Optically they look like stars. Quasars are the active nuclei of Seyfert galaxies too far away to see the host galaxy.

In 1943 Carl Keenan Seyfert discovered a class of galaxies with nuclei emitting radiation from highly ionized gas with spectra containing strong emission lines. Seyfert galaxies have located at their centres a supermassive black hole with a mass lieing in the range one million to one billion solar masses. There are two main types of Seyfert galaxies known as type 1 and type 2. A type 1 Seyfert galaxy has a spectrum that has both broad and narrow emission lines. A type 2 Seyfert galaxy has a spectrum that has only narrow emission lines. At present they are given a fractional classification applying the relative intensities of the broad and narrow emission lines, for instance, type 1.5 or type 1.9. My MPhil examines all types of AGN with a view to understanding their connection with star formation in their host galaxies.

It is thought that all Seyfert galaxies are of the same type, and only appear to be different from the direction of viewing. This is known as the Seyfert unification theory. In type 2 Seyfert galaxies it is believed that the broad emission lines are obscured by dust, gas and the angle of viewing on the galaxy. The main effect will depend on the galaxy.

All galaxies have at their centre a supermassive black hole and they are apparently linked from birth. The central supermassive black hole's mass is roughly half a percent of the total mass of the host galaxy, (Holland et al. 1998).

A quasar's power source is the conversion of gravitational potential energy in to heat, in the form of gas temperature and infrared radiation, by matter falling on to the central supermassive black hole via an accretion disc. The accretion disc is a swirling collection of gas and dust falling on to the central supermassive black hole, which gets very hot by friction, hence it emits huge amounts of electromagnetic radiation from radio to X-rays. There are some radio quiet and radio loud quasars. As the matter falls in to the central supermassive black hole it crosses the event horizon and disappears. The accretion disc is useually ten to ten thousand times the size of the Schwarzschild radius of the central supermassive black hole. The Schwarzschild radius is the radius of the event horizon.

QSO stands for, Quasi Stellar Object, which is another name for a quasar.

LIRG, stands for, Luminous Infra Red Galaxy, which is a galaxy type with high infrared emission. The infrared luminosity of a LIRG is in the range, 10^{11}-10^{12}, times the solar luminosity. But some LIRGs are AGNs.

ULIRG, stands for Ultra Luminous Infra Red Galaxy, which is a galaxy type, with strong infrared emission. The infrared luminosity of a ULIRG is in the range, 10^{12}-10^{13}, times the luminosity of the sun. But some ULIRGs are AGNs.

HLIRG, stands for Hyper-Luminous Infra Red Galaxy, which is a galaxy type with enormous infrared emission. The infrared luminosity of a HLIRG is larger than, 10^{13} times the luminosity of the sun. But some HLIRGs are AGNs.

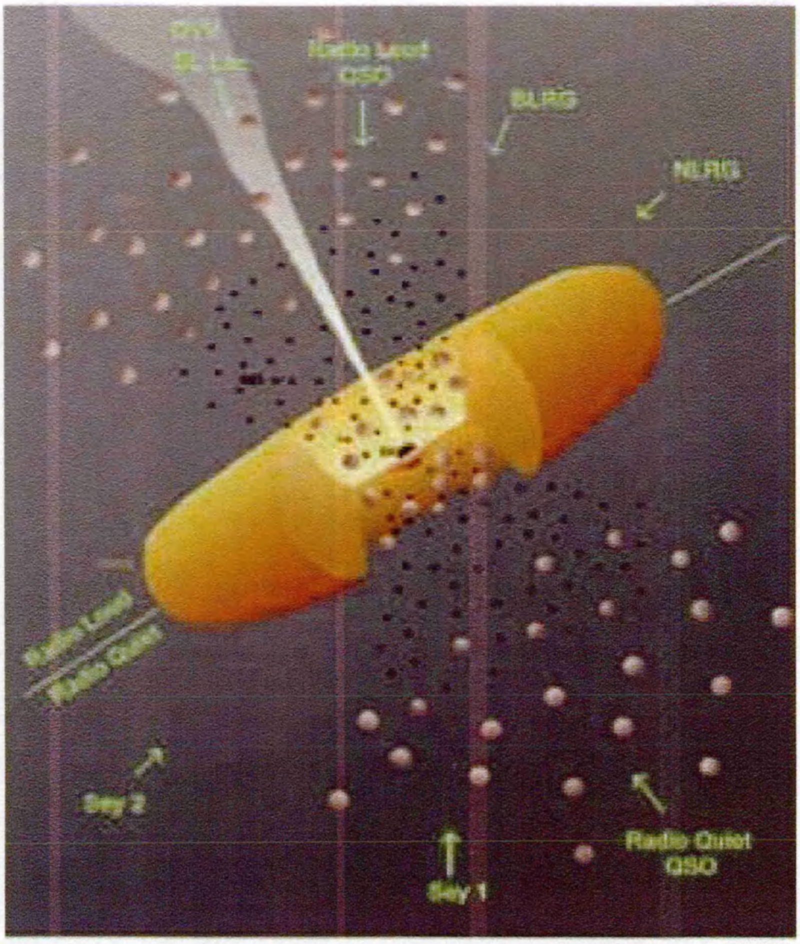


Figure 1.2: This is the diagram of a general AGN from Urry & Padovani 1995. This diagram shows the central engine, accretion discs and the gas and dust torus. It also indicates the various types of AGN, and their dependence on viewing angle. Top names are radio loud and bottom are radio quiet.

Blazar, stands for blazing quasi-stellar object. This is a type of quasar. Blazars are very compact quasars that are linked to supermassive black holes lying in the centre of giant elliptical galaxies. Blazars are second only to Gamma Ray Bursters, GRBs, in being the most energetic objects known in the Universe. There are two types of blazars, such as: OVV, Optically Violently Variable and BL Lac objects. Blazars are AGN having a relativistic jet that is pointing in our direction.

BLRG, which stands for Broad Line Radio Galaxy, this is another type of AGN. BLRG are type 1 Seyfert galaxies.

In 1974, B.L. Fanaroff and J.M.Riley discovered, that the relative positions of areas of strong and weak surface brightness in the radio lobes of radio galaxies were linked with their radio luminosity. They separated the radio galaxies they observed into two classes. They used pf the ratio R_{FR} of the distance between the areas of maximum surface brightness that are located on opposite sides of the galaxy or quasar at the centre, to the size of the source up to the minimum brightness contour in the map. If $R_{FR} < 0.5$ the source is put in to class I. If $R_{FR} > 0.5$ the source is put in to class II.

NLRG, stands for Narrow Line Region Galaxy, this an AGN type, that are type 2 Seyfert galaxies.

1.4 Feedback

1.4.1 What is feedback?

The redshifts of quasars seen currently have a maximum of 7.1, so some quasars are very distant objects, Matlock et al. 2011. It is thought that there may be a link between quasars black hole accretion and star formation in their host galaxies. Quasars are feeding supermassive black holes. As quasars feed they grow in size, so in time their accretion discs get so hot that their radiation and gas pressure gets strong enough to push the gas and dust away. Hence the supermassive black hole stops feeding and the quasar turns itself off. The gas and dust is expelled from the supermassive black hole, causing shock waves in the surrounding galaxy, which could trigger more star formation. This leads to a chain reaction of supernovae and star formation. Alternatively the wind from the central engine could expel so much gas it suppresses star formation. There is evidence in local galaxies that the accretion of matter on to supermassive black hole causing them to grow is strongly linked to the growth of the galaxies with these black holes at their centre. The black hole masses correlate strongly with their host galaxy properties, see Magorrian et al. 1998 for details, despite the black hole's mass being half a percent or so of the total mass of the host galaxy. The black hole's size is a very tiny fraction of the total galaxy volume. These phenomena all form part of the feedback process between a quasar and the host galaxy. This is the topic of my MPhil.

1.4.2 The determination of the central super massive black hole mass

The most direct means for measuring the masses of central supermassive black holes is the stellar proper motion method. Let us consider how we could get the mass of the supermassive black hole lying at the centre of the Milky way Galaxy. The distance between the galactic centre and Earth is roughly 8kpc. Hence one parsec seen at that distance would be same as the angle of about 25 arc seconds. This angle is big enough to alow the tracing of the orbital paths of distinct stars near the galactic centre, and obtain direct mass measurements. Due to the high level of dust between us and the galactic centre, we must use the near infrared. Ground based telescopes using adaptive optics using the wavelength range, 1-3μm, that have primary mirror diameters in the range, 8-10m, can give results for long observing runs that are as good as the HST. The mass inside a given star's orbit near to the galactic centre can be deduced from the size and shape of the orbit and the star's orbital period. The mass enclosed inside a stellar orbit, can then be used to obtain the enclosed mass as a function of distance from the galactic centre. The best current mass for the central supermassive black hole mass is, $(4.5 \pm 0.4) \times 10^6$, solar masses, see Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details.

Reverberation mapping is another method to get estimates of the size and mass of a supermassive black hole. This approach applies the variability of the light generated by the accretion disc lying immediately outside the feeding supermassive black hole. If an object's luminosity varies over time scales of weeks, then it is likely that its size is that of light-weeks or less. Quasar luminosities are known to change over periods in the range of years to days. The very high luminosities of quasars, the small sizes of quasars deduced from reverberation mapping, and the very large radiative efficiency of matter being accreted on to a black hole lead to the idea that power sources of quasars are supermassive black holes, see Stephen Serjeant's book, observational cosmology, Cambridge 2010, for details.

The quasar can change luminosity with time because of light travel time effects or relativistic beaming, or both. A refinement of this method is known as reverberation mapping, which eliminates the effects of relativistic beaming. A flash emission can produce an increase in the brightness of the broad emission lines, but there will be a time lag from due to the broad emission lines being not located at the centre of the quasar. Plus, the clouds will have different physical properties than are present in the relativistic jet. By studying the changes in the continuum and comparing them with the changes in the emission lines, you can deduce how close the emission line clouds are to the central bright engine of the quasar. The flux of the emission lines is directly proportional to the cross-correlation, $\Sigma C(t)L(t - \Delta t)$ where *C(t)* are the continuum flux measurements and *L(t)*, are the line flux measurements. The time delay is obtained by connecting the line and continuum observations, that is getting the total of $\Sigma C(t)L(t - \Delta t)$ for a number of calculated lags Δt.

One explanation for why the lag measurements are quite wide in some cases, is that if Δt is given a value very near to but not the same as the actual value, then it is possible that there is an amount of positive correlation with the signal. More importantly, the light-travel time effects imply that the broad line clouds are sampled over a range of radii for any given fixed time lag. The reverberation mapping method can be used for distinct emission lines in the spectra which may be used to deduce the internal arrangement of the broad emission line regions of galaxies with AGN. High ionization lines are the fastest in responding, suggesting ionization layering, and these lines are the broadest emission lines. This clearly implies a method to deduce an estimate of the mass of the central supermassive black hole: if the widths of the emission lines are produced by the Doppler effect, then matter nearer to the central supermassive black hole is moving faster, then the black hole mass M_{BH} is deduced by the equation:

$$M_{BH} \simeq \frac{r(\Delta v)^2}{G} \tag{1.23}$$

where r is the radius deduced from the reverberation measurements and Δv is obtained from the emission line widths.

1.4.3 The Magorrian relation

There is no strong link between the total mass of the host galaxy and the mass of the central supermassive black hole. There is a stronger and more useful link between M_{BH} the mass of the central supermassive black hole, and the brightness of the spheroidal part of the galaxy. For a spiral galaxy the spheroidal part of the galaxy is the galactic bulge. For an elliptical galaxy, the spheroidal part of the galaxy is the whole galaxy.

The link described above was first seen by John Magorrian and his team in 1998. Since then it is known as the Magorrian relation. The Magorrian et al. 1998 paper explained how they set up dynamical models for a sample of thirty six local galaxies using the HST photometry and ground-based kinematics. Their models makes four assumptions. The first assumption is that each individual galaxy is axisymmetric.

The second assumption is that each galaxy has a two-integral distribution function. The third assumption is that each galaxy has an arbitrary inclination angle. The fourth assumption is that each galaxy has a position independent mass-to-light ratio Υ and a central massive dark object, MDO, of arbitrary mass $M_\bullet$. The Magorrian et al. 1998 paper gives good fits for thirty two of the sample of galaxies, ausing some value of $M_\bullet$ and Υ. The other four galaxies in their sample that can not be fitted have kinematically decoupled cores. The mass-to-light ratios used for the thirty two galaxies that are well-fitted by the model, are in agreement with the fundamental plane correlation $\Upsilon \propto L^{0.2}$ where L is galaxy luminosity. With the exception of only six galaxies, the models produce a 95% confidence level in an estimate of the MDO of mass $M_\bullet \sim 0.006 M_{bulge} \equiv 0.006 \Upsilon L$. Five of the six galaxies that do not fit with the model show that $M_\bullet = 0$ using this value of $M_\bullet$ they are consistent with this correlation. The last of the six misfits, NGC7332, has a much stronger upper limit on $M_\bullet$.

1.4.4 The Ferrarese and Merritt relation

There is a much stronger link between the speed of the stars that make up the galaxy spheroid σ and the mass of the central supermassive black hole M_{BH}. This link is known as the Ferrarese and Meritt relation. The spread of observational data points that form the data set is almost entirely due to observational error, this means the fundamental scatter in measurement is most likely zero. The equation of best-fit for the data is

$$M_{BH} = (1.66 \pm 0.32) \times 10^8 M_\odot \left(\frac{\sigma}{200 km/s}\right)^{4.58 \pm 0.52} . \tag{1.24}$$

Chapter 2

Herschel

2.1 Preview of chapter two

In chapter two the Herschel space observatory and the two instruments I shall be amalysing data from, PACS and SPIRE, are described. Plus the HATLAS project is discussed. Data from PACS and SPIRE, from the early science phase, will be used to give evidence for star formation in my list of Quasar candidates. This will be discussed in chapter four.

2.2 The Herschel space observatory

We do not know if all galaxies were Seyfert galaxies at an early stage in their evolution. We also do not know if all quasars are the same type, or the variation in behavior is an evolutionary sequence. In order to establish a link between quasars and star formation in their host galaxies, I will need to gather a lot of observational data from various sources, such as Herschel.

Herschel is an infrared astronomical telescope on a satellite at a distance of 1.5 million kilometres from Earth, Pilbratt et al. 2010. There are four instruments on Herschel. I will only use two of them PACS and SPIRE. PACS stands for Photo detector Array Camera and Spectrometer, Poglitsch et al. 2010. SPIRE stands for Spectral and Photometric Imaging Receiver, Griffin et al. 2010. The main mirror diameter is 3.5 metres, which makes it the largest space telescope. The primary mirror is bigger than that of the Hubble Space Telescope, HST. The PACS instrument works at 70, 100 and 160 microns. We only have access to the 100 and 160 micron data because the PACS instrument only works on two wave bands at once. These two wave bands are useful to us because they can be used for detecting star formation, because star formation peaks at rest frame wavelengths of about 60 to 80 microns, and is hence redshifted into the PACS wavelengths. The SPIRE instrument works at 250, 350 and 500 microns. The SPIRE instrument is more sensitive than the PACS instrument, which is why sources seen in PACS but not in SPIRE are interesting and perhaps unexpected.

2.2.1 PACS

PACS, is an imaging camera and a low-resolution spectrometer using the wavelength bands given in the previous section. The spectrometer has a spectral resolution lying in the range $R = 1000$ to $R = 5000$, and can detect signals as low as about 100mJy. It functions as an integral field spectrometer, using both spatial and spectral resolution. The PACS instrument can only function with two wave bands at once, either 60-85 microns, 85-130 microns and 130-210 microns. The PACS camera can only detect sources with fluxes greater than a couple of millijanskys.

2.2.2 SPIRE

SPIRE, is an imaging camera plus a low-resolution spectrometer using the wavelength range of 194 to 672 microns. The spectrometer has a resolution lying in the range $R = 40$ to $R = 1000$ at the wavelength of 250 microns. The spectrometer can detect point sources with a flux larger than about $100mJy$, and detect extended sources with a flux larger than about $500mJy$ at 5σ. The camera uses three bands, centered on 250, 350 and 500 microns, with pixel widths of 139, 88 and 43 pixels respectively. The SPIRE camera can detect point sources with fluxes greater than 2 mJy. The SPIRE camera can detect extended sources with fluxes between 4mJy and 9mJy.

2.3 ATLAS

ATLAS stands for Astrophysical Terahertz Large Area Survey. The Herschel ATLAS is the largest Open Time Key Project, OTKP, that will use the Herschel satellite. The survey will cover an area of the extragalactic sky with an area of 550 square degrees, which is four times bigger than the sum total of sky coverage of all the other Herschel surveys. It will take measurements in five submillimetre and far infrared wave bands, (Eales et al. 2009).

2.3.1 The motivation for the Herschel space telescope

About half the energy emitted by all object types that have existed since the big bang has been absorbed by dust and been re-radiated in the wavelength range, 60μm to 500μm, Dwek et al. 1998, Fixsen et al. 1998 and Driver et al. 2008. This wavelength range is still currently relatively poorly studied, relative to other infrared and submillimetre wave bands. In the 1980s the Infrared Astronomical Satellite, IRAS, made a survey of the whole sky at the wavelengths of 60μm and 100μm. But, virtually in every case of the sample of tens of thousands of galaxies seen in the IRAS survey were starburst spiral galaxies, lying in the local universe. This means they had redshifts less than, 0.1. Sadly, IRAS gave little data about the dust content of other galaxy types, Bregman et al. 1998. Even for the case of galaxies of late Hubble type only a small fraction of the dust inside the galaxy was hot enough to emit strongly enough in the far infrared to be seen in the IRAS data. Spitzer and ISO made use of the longest wavelength $\sim$ 170μm range, these instrument surveys had less problems in this area. But they still did not detect any dust with a temperature of less than, 15K, Bendo et al. 2003. Despite the successes achieved by Spitzer, ISO, and IRAS, the majority of the wavelength range, 60μm to 500μm, is still currently poorly studied.

For the purposes of the Herschel space observatory instruments, the submillimetre waveband is defined to be the wavelength range, 100μm to 1mm. The lack of a survey of a large fraction of the submillimetre sky, has led to the situation that we have a much better understanding of some of the properties of dust in the early cosmos, than those same properties of the dust that is present in the cosmos today.

2.3.2 The reason for the H-ATLAS survey

The Herschel satellite was launched by the ESA on 14th May 2009. Herschel will help improve our understanding of dust in general plus dust hidden star formation, especially in the local cosmos.

The H-ATLAS project has been given six hundred hours of observing time on Herschel. This makes it the biggest key project that is being run in Herschel Open Time. In all of its science aims, the end sensitivity is not as important to the survey as the total area of coverage of the sky by the survey. Hence it is using the fastest scan rate of the telescope has, 60 arc-seconds per second. For the first science run, H-ATLAS will make measurements using both the PACS and SPIRE instruments. Hence it will use the Herschel observing mode that allows the telescope to make simultaneous measurements for both the PACS and SPIRE instruments. This is known as the Parallel Mode, or P Mode. One takes measurements at the wavelengths of 100μm and 160μm, and the other takes measurements at the wavelengths of 70μm and 160μm. The H-ATLAS survey chose to use the PACS configuration that uses 100μm and 160μm. This choice was made mainly for sensitivity reasons.

Considering the subsequent uses of the H-ATLAS data, the survey team made the choice of observing areas that lie in the southern hemisphere, the northern hemisphere and the equator of the celestial sphere. Plus, the team chose the observing areas to get the biggest amount of extra data, and minimize the noise due to dust emissions in our own Galaxy. The choices of viewing fields where made using galactic dust emission data from the IRAS 100μm maps.

One of the fields is located near the north galactic pole with an area of one hundred and fifty square degrees. This field is known as the NGP field. There are three fields, each having an area of thirty six square degrees lying in the same region of the sky as the Galaxy And Mass Assembly (GAMA) redshift survey, Driver et al. 2009. These fields are known as the GAMA fields. Lastly there are two fields with an overall total area of two hundred and fifty square degrees, located near the south galactic pole. These fields are known as the SGP fields.

The total area of sky coverage of all the fields that make up the survey is five hundred and ten square degrees. The angular resolution, the full width at half maximum, of the observations will be roughly, 7 arc seconds at a wavelength of 100 microns, 12 arc seconds at a wavelength of 160 microns, 18 arc seconds at a wavelength of 250 microns, 25 arc seconds at a wavelength of 350 microns, and, 36 arc seconds at a wavelength of 500 microns. The 5σ detection limits that should be achieved in the five wave bands are, 67mJy at 100μm, 94mJy at 160μm, 45mJy at 250μm, 62mJy at 350μm, and, 53mJy at 500μm, Eales et al. 2009. These values are right for sources that are not near by extended sources such as galaxies. Plus, we have assumed that there is negligible dust emission from within our own Galaxy, and there is no confusion with sources that are outside our own Galaxy.

2.3.3 The summary of the H-ATLAS survey

There are six working groups that are executing studies as part of the H-ATLAS survey, Eales et al. 2010. There are many other extra projects possible in H-ATLAS. The wavelength range 200μm to 500μm, used in H-ATLAS, is still mainly unexplored. There is a high probability that H-ATLAS will make a number of unexpected discoveries.

It is hoped that H-ATLAS will detect a sample of about forty thousand galaxies lying in the nearly local Cosmos, this means that they will have redshifts less than 0.3. The majority of this sample of galaxies will have redshifts less than 0.1.

2.3.4 The contribution the Planck HFI instrument to the H-ATLAS survey

The Planck HFI which stands for Planck High-Frequency Instrument, is going to be used to make a survey of the whole sky at the following six wavelengths, 3mm, 2.1mm, 1.4mm, 850μm, 550μm, and 350μm, Lamarre J-M. et al. 2003. This will be the first survey of the whole sky made using these wavelengths. Plus, the observations of the CMB by the HFI instrument there will be measurements made of the emission of the extended foreground, for example the dust in our own Galaxy. There is a good chance that the survey will detect a large number of point like sources, de Zotti et al. 2005.

2.3.5 The H-ATLAS survey's contribution to detecting gravitational lenses

It is theoretically possible, that gravitational lensing can be used for studying the time evolution of the mass distribution of galaxies. This is a vital test of models of galaxy structure formation. In practice, it is very difficult to get a big enough sample of gravitational lenses. Currently the largest survey of gravitational lenses looked for lenses in radio sources with flat-spectra. But, the data from high-resolution radio searches, only found 22 gravitational lenses out of a sample of 16000 radio sources, Browne 2003.

2.3.6 The contribution of the H-ATLAS survey to detecting supermassive black holes

Surveys made at submillimetre wavelengths that have been executed at large redshift, have seen about five to ten percent of their sources are quasars, Priddley et al. 2003, and Beelen et al. 2006. The spectra of these quasars give evidence that most of the energy emitted at submillimetre wavelengths is produced by starburst surrounding the active galactic nucleus and not from the galactic nucleus itself. This is evidence that the formation of the central supermassive black hole and star formation in the host galaxy are linked, see section 1.3.

2.3.7 The H-ATLAS survey's contribution to detecting AGNs

H-ATLAS will be used to see the individual galaxies that have active galactic nuclei, that are lying in the H-ATLAS area. Then a stacking analysis (coadding) of the AGN will be done. This approach will allow sources to be seen that are too dim to be seen without this approach. Next a statistical analysis of the AGN properties will be done, hence a good distribution of their properties will be deduced.

Chapter 3

Data analysis

3.1 Preview of chapter three

In chapter three, I will derive the Vio kernel and discuss the typos in the Vio et al paper. Plus I will derive the modified kernel for filtering and discuss the SNR penalty for white noise. I will outline the three versions of the PACS pipeline and described the calibration of the first and third versions of the PACS pipeline. Lastly I will explain the timeline filtering and cross-scan filtering used in the third version of the PACS pipeline.

3.2 Theoretical basis

3.2.1 The Vio analytic analysis

The problem I am trying to solve is to find the location of point sources along with their fluxes in the submillimetre wave band. I will also gather data from the H-

ATLAS survey for links between quasars and star formation in their host galaxies. Vio et al. 2002 first obtained a suitable kernel which maximises the signal to noise. They filter the input signal by convolving it with the kernel they have derived. The Fourier transform of the kernel is directly proportional to the Fourier transform of the Point Spread Function, PSF, divided by the power frequency spectrum of the noise. The kernel is obtained by taking the inverse Fourier transform of this. The constant of proportionality is derived later on in this section.

The SCUBA matched filter technique outlined in Serjeant et al. 2003 starts by generating a new variable W, where

$$W = \frac{1}{N^2} \tag{3.1}$$

and N is the noise. The SCUBA matched filter does the filtering by convolving the input signal multiplied by W with the PSF and then dividing by W convolved with the PSF squared.

Vio et al. presented an ostensibly optimal point-source finding algorithm in the presence of a Gaussian random field as a background. I have shown the main results in the Vio et al. source extraction analysis are correct, though I found a typo in the paper, and verified that the matched filtering expressions used in earlier submillimetre wave surveys are also correct. I have managed to obtain the kernel in the Vio et al. paper I need to do the filtering by the calculus of variations with undetermined Lagrangian multipliers. This kernel gives the maximum signal to noise. The Vio et al. paper quotes the Gauss-Markov theorem to get this result. This proved to be a dead end using the forms I found of this theorem. I was able to prove the Gauss-Markov theorem in several different ways, but when I applied it to this problem it did not constrain the filter. However, my alternative proof recovered the Vio result.

The SCUBA filter assumes a white noise background but an RMS that varies from position to position. This variation is uncorrelated with the signal.

The Vio filter assumes a constant white noise background with a fixed RMS but adds a structure with a known power spectrum.

We are operating under the assumption that if we have the general case of structured background with a known power spectrum, plus a white noise component, its own RMS varying with position but not correlated with the signal, then the optimal filter is the SCUBA filter with PSF replaced by the Vio kernel.

The Vio et al. 2002 paper therefore gives us the kernel, we need to use with SCUBA matched filter to do the filtering on the PACS data.

I now give a more detailed summary of the Vio et al. 2002 paper on point source filtering. In the timeline filtering part of the PACS pipeline we will be working in one dimension, the time dimension. I also discuss an earlier approach in this thesis. In the earlier approach to the spatial filtering of the PACS pipeline we worked in two dimensions, right ascension α and declination δ on the sky. I will discuss this in section 3.2.1. In the Vio et al. 2002 paper they work in n dimensions, where n is any integer greater than or equal to one.

The sources are assumed to be point-like with a given shape, $\tau(\overrightarrow{x})$. $\overrightarrow{x}$ is the vector in n dimensions describing the location of a point in space, y is the flux level of the wavelength we are interested in, which is a function of, $\overrightarrow{x}$, $y(\overrightarrow{x})$. $\overrightarrow{x} \in R^n$. The expression connecting, $\overrightarrow{x}$ and y, is, $y(\overrightarrow{x}) = \sum_{j=1}^{N} S_j(\overrightarrow{x}) + z(\overrightarrow{x})$, where N is the number of sources in the field, $S_j(\overrightarrow{x}) = A_j \tau(\overrightarrow{x} - \overrightarrow{x_j})$, A_j is the amplitude of source j, $\overrightarrow{x_j}$ is the location of source j, and $z(\overrightarrow{x})$ is the background with zero mean and power-spectrum $P(\vec{q})$. A_j and $\vec{x_j}$ are unknown. $\vec{q}$ is the n dimensional Fourier space variable. $\vec{q}$ in general will be a n dimensional complex variable. $P(\vec{q})$, is given below.

$$E[z(\vec{q})z^*(\vec{q'})] = P(\vec{q})\delta^n(\vec{q} - \vec{q'}) \tag{3.2}$$

E[.], denotes the expectation value of the quantity in the square brackets. *, as a super script means the complex conjugate of the quantity it is used as a super script to. $\delta^n(\vec{q} - \vec{q'})$ is the n dimensional Dirac distribution. $z(\vec{q})$ is the Fourier transform of $z(\vec{x})$ defined as,

$$z(\vec{q}) = \int_{-\infty}^{\infty} z(\vec{x})e^{-i(\vec{q}.\vec{x})}d\vec{x}. \tag{3.3}$$

To find point sources we need to obtain good estimates of the source locations $\vec{x_j}$ and the amplitudes or fluxes of the sources A_j. A traditional technique for doing this is to locate peaks in the cross-correlation function.

$$c(\vec{x}) = \int_{-\infty}^{\infty} y(\vec{x} + \vec{b})\tau(\vec{b})d\vec{b}. \tag{3.4}$$

The approach is that $c(\vec{x_0})$ detects the closeness between the source shape with a part of $y(\vec{x})$ centered on $\vec{x_0}$; equation 3.4, is acting like a filter of the signal $y(\vec{x})$ with a filter $\tau(\vec{b})$ which enhances the frequencies that are typical of the source. As soon as the locations of the sources are known, their amplitudes can be estimated by using a fitting technique such as least squares.

The cross-correlation function, equation 3.4, does not use any of the background noise properties. If the power spectrum of the background noise $P(\vec{q})$ is known or can be well estimated, other techniques that make use of this information can be applied. Such procedures are the main thrust of the Vio et al. 2002 paper.

The procedure that is outlined in the Vio et al. 2002 paper, is given here. The signal is filtered to bring out the sources relative to the background. This is carried out by cross-correlating the signal $y(\vec{x})$ with a filter Φ as in equation 3.4 replacing τ by Φ. The locations of the sources are then identified by obtaining the peaks in the filtered signal that have peaks greater than a given minimum value. The source strengths are obtained by taking the values of the peaks for each source.

The shape of the source profile of the sources we are interested in is known. The background noise power spectrum is known. The source shape is spherically symmetric, specified by a scale R_S, and the background noise is assumed to be isotropic. In these circumstances we can set $s(\vec{x}) \equiv s(x)$, $x = ||\vec{x}||$ and $P(\vec{q}) \equiv P(q)$, where $q = ||\vec{q}||$. We also assume that there is negligible source overlap.

We will now look at the collection of spherically symmetric filters $\Phi(\vec{x};\vec{b})$, such as, $\Phi(\vec{x};\vec{b}) = \phi(||\vec{x} - \vec{b}||)$, with its Fourier transform $\phi(q)$. $w(\vec{b};\phi)$ is the filtered field.

$$w(\vec{b};\phi) = \int_{-\infty}^{\infty} y(\vec{x})\Phi(\vec{x};\vec{b})d\vec{x} \qquad (3.5)$$

$$w(\vec{b};\phi) = \int_{-\infty}^{\infty} y(\vec{q})\phi(q)e^{i(\vec{q}.\vec{b})}d\vec{q} \qquad (3.6)$$

$y(\vec{q})$ is the Fourier transform of $y(\vec{x})$. $\phi(q)$ is the Fourier transform of $\phi(\vec{x})$. $s(q)$ is the Fourier transform of $s(x)$. The mean of $w(\vec{b};\phi)$ is given by $\mu(\vec{b};\phi)$ below after the multiple integrals have been converted to a single integral by changing to n dimensional spherical polar coordinates and doing the integrals that can be done.

$$\mu(\vec{b};\phi) = E[w(\vec{b};\phi)] \qquad (3.7)$$

$$\mu(\vec{b};\phi) = \alpha\left(\int_{0}^{\infty} q^{n-1}s(q)\phi(q)e^{i(\vec{q}.\vec{b})}dq\right) \qquad (3.8)$$

The variance of $w(\overrightarrow{b};\phi)$ is given by $\sigma^2(\phi)$ below after a similar change in coordinates as used with $w(\overrightarrow{b};\phi)$ has been applied.

$$\sigma^2(\phi) = E[w^2(\overrightarrow{b};\phi)] - \mu^2(\overrightarrow{b};\phi) \tag{3.9}$$

$$\sigma^2(\phi) = \alpha\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right) \tag{3.10}$$

α is given by $\alpha = 2\pi^{n/2}/\Gamma(n/2)$.

The first restriction on the filter is from the second part of the approach; the locations of the sources are assumed to be known and the task is to obtain the amplitudes. Given the assumed location of the sources, it is sufficient to consider a field $y(\overrightarrow{x})$, as specified in the general definition at the beginning of this section, so we take a single source at the origin $s(\overrightarrow{x}) = A\tau(\overrightarrow{x})$. To obtain A we have to make $w(\overrightarrow{0};\phi)$ an unbiased estimator of A, that is $\mu(\overrightarrow{0};\phi) = A$. Hence ϕ must satisfy the constraint

$$\int_0^\infty q^{n-1}\tau(q)\phi(q)dq = \frac{1}{\alpha}. \tag{3.11}$$

To strengthen the magnitude of the source with respect to the background we need to specify the filter Φ that minimizes the variance $\sigma^2(\phi)$. This is another way of saying that the signal to noise at the origin $D(\phi)$ is a maximum. $D(\phi)$ is a good means of determining whether a source will be detected, for a source of known location. This is also works in the general case.

$$D(\phi) = \frac{\mu(\overrightarrow{0};\phi)}{\sigma(\phi)} \tag{3.12}$$

Φ is specified in such a way that $w(\overrightarrow{0};\phi)$ is a minimum variance unbiased estimator of A, which is linear in $y(\overrightarrow{x})$. By applying the Gauss-Markov theorem, Vio et al.

claim that the generalized least squares estimator of A is given by the filter

$$\phi(q) = \frac{\tau(q)}{\alpha a P(q)}, \tag{3.13}$$

$$a \equiv \int_0^\infty \frac{q^{n-1}\tau^2(q)dq}{P(q)} \tag{3.14}$$

with minimum variance,

$$\sigma^2(\phi) = \frac{1}{\alpha a}. \tag{3.15}$$

I will now briefly outline how I used the calculus of variations with Lagrangian undetermined multipliers to obtain the Vio kernel ϕ. Putting $\overrightarrow{b} = \overrightarrow{0}$ in equation 3.8 I obtain,

$$\mu(\overrightarrow{0};\phi) = \alpha\left(\int_0^\infty q^{n-1}s(q)\phi(q)dq\right). \tag{3.16}$$

Then I take the square root of equation 3.10, to obtain

$$\sigma(\phi) = \alpha^{\frac{1}{2}}\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)^{\frac{1}{2}}. \tag{3.17}$$

Transforming $s(\overrightarrow{x}) = A\tau(\overrightarrow{x})$ to Fourier space variables and using n dimensional spherical polar coordinates, I obtain,

$$s(q) = A\tau(q). \tag{3.18}$$

Using equation 3.12, substituting in for $\mu(\overrightarrow{0};\phi)$, $\sigma(\phi)$, and then replacing, *s(q)*, with the form given in equation 3.18, I obtain,

$$D(\phi) = \frac{A\alpha^{\frac{1}{2}}\left(\int_0^\infty q^{n-1}\tau(q)\phi(q)dq\right)}{\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)^{\frac{1}{2}}}. \tag{3.19}$$

Re-arranging equation 3.11, this gives,

$$\int_0^\infty q^{n-1}\tau(q)\phi(q)dq - \frac{1}{\alpha} = 0. \tag{3.20}$$

$D(\phi)$ is an example of a functional. That is an operator that maps a function such as ϕ to a real number. This allows us to maximise $D(\phi)$. Equation 3.20 gives us an additional constraint for ϕ to satisfy. Hence we need to use Lagrangian undetermined multipliers to maxmise $D(\phi)$. We need to maxmise the new functional $\overline{D(\phi)}$, which is given below,

$$\overline{D}(\phi) = D(\phi) - \lambda\left(\int_0^\infty q^{n-1}\tau(q)\phi(q)dq - \frac{1}{\alpha}\right) \tag{3.21}$$

where λ is the Lagrangian undetermined multiplier. $\Delta\overline{D}[\phi, h]$, is the Gateaux differential of $\overline{D}(\phi)$, where h is the admissible path.

$$\Delta\overline{D}[\phi, h] = \left(\frac{d}{d\epsilon}\left(\overline{D}(\phi + \epsilon h)\right)\right)_{\epsilon=0}. \tag{3.22}$$

Since the limits of the integral in equation 3.16 and equation 3.17 are fixed the admissible path must satisfy $h(0) = 0$ and $h(\infty) = 0$. The Gateaux differential of $\overline{D}(\phi)$ becomes,

$$\Delta\overline{D}[\phi, h] = A\alpha^{\frac{1}{2}}\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)^{\frac{-3}{2}}(T_1 - T_2 - T_3). \tag{3.23}$$

$$T_1 = \left(\int_0^\infty q^{n-1}\tau(q)h(q)dq\right)\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right) \tag{3.24}$$

$$T_2 = \left(\int_0^\infty q^{n-1}\tau(q)\phi(q)dq\right)\left(\int_0^\infty q^{n-1}P(q)\phi(q)h(q)dq\right) \tag{3.25}$$

$$T_3 = \left(\frac{\lambda}{A\alpha^{\frac{1}{2}}}\right)\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)^{\frac{3}{2}}\left(\int_0^\infty q^{n-1}\tau(q)h(q)dq\right) \tag{3.26}$$

For stationary paths,

$$\Delta\overline{D}[\phi, h] = 0. \tag{3.27}$$

Substituting for the Gateaux differential in equation 3.27 and simplifying I find,

$$T_4 - T_5 - T_6 = 0. \tag{3.28}$$

$$T_4 = \left(\int_0^\infty q^{n-1}\tau(q)h(q)dq\right)\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right) \tag{3.29}$$

$$T_5 = \left(\int_0^\infty q^{n-1}\tau(q)\phi(q)dq\right)\left(\int_0^\infty q^{n-1}P(q)\phi(q)h(q)dq\right) \tag{3.30}$$

$$T_6 = \left(\frac{\lambda}{A\alpha^{\frac{1}{2}}}\right)\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)^{\frac{3}{2}}\left(\int_0^\infty q^{n-1}\tau(q)h(q)dq\right) \tag{3.31}$$

It is useful to define the variable a_0.

$$a_0 = \int_0^\infty q^{n-1}P(q)\phi^2(q)dq \tag{3.32}$$

Combining equations 3.28 and 3.32 I obtain,

$$T_7 - T_8 - T_9 = 0. \tag{3.33}$$

$$T_7 = \left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)\left(\int_0^\infty q^{n-1}\tau(q)h(q)dq\right) \tag{3.34}$$

$$T_8 = \left(\int_0^\infty q^{n-1}\tau(q)\phi(q)dq\right)\left(\int_0^\infty q^{n-1}P(q)\phi(q)h(q)dq\right) \tag{3.35}$$

$$T_9 = \left(\frac{\lambda a_0^{\frac{1}{2}}}{A\alpha^{\frac{1}{2}}}\right)\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)\left(\int_0^\infty q^{n-1}\tau(q)h(q)dq\right) \tag{3.36}$$

Transforming equation 3.33 to a double integral and simplifying I find,

$$\int_0^\infty\int_0^\infty q^{n-1}Q^{n-1}\left(P(q)\phi^2(q)\tau(Q) - \tau(q)\phi(q)P(Q)\phi(Q) - \frac{\lambda a_0^{\frac{1}{2}}P(q)\phi^2(q)\tau(Q)}{A\alpha^{\frac{1}{2}}}\right)h(Q)dqdQ = 0. \tag{3.37}$$

By the fundamental lemma of the calculus of variations, I obtain,

$$q^{n-1}Q^{n-1}\left(P(q)\phi^2(q)\tau(Q) - \tau(q)\phi(q)P(Q)\phi(Q) - \frac{\lambda a_0^{\frac{1}{2}}P(q)\phi^2(q)\tau(Q)}{A\alpha^{\frac{1}{2}}}\right) = 0. \tag{3.38}$$

On solving equation 3.38 for $\phi(Q)$, I obtain

$$\phi(Q) = \frac{(A\alpha^{\frac{1}{2}} - \lambda a_0^{\frac{1}{2}})P(q)\phi(q)\tau(Q)}{A\alpha^{\frac{1}{2}}\tau(q)P(Q)}. \tag{3.39}$$

On rearranging equation 3.39 I get,

$$\phi(q) = \frac{A\alpha^{\frac{1}{2}}P(Q)\tau(q)\phi(Q)}{(A\alpha^{\frac{1}{2}} - \lambda a_0^{\frac{1}{2}})P(q)\tau(Q)}. \qquad (3.40)$$

Substituting for $\phi(q)$ in equation 3.20 I reach,

$$\int_0^{\infty} \tau(q)\left(\frac{A\alpha^{\frac{1}{2}}P(Q)\tau(q)\phi(Q)}{(A\alpha^{\frac{1}{2}} - \lambda a_0^{\frac{1}{2}})P(q)\tau(Q)}\right) dq - \frac{1}{\alpha} = 0. \qquad (3.41)$$

Solving equation 3.41 for $\phi(Q)$, replacing Q by q, and using equation 3.14, the result is,

$$\phi(q) = \left(\frac{(A\alpha^{\frac{1}{2}} - \lambda a_0^{\frac{1}{2}})}{A\alpha^{\frac{3}{2}}a}\right)\left(\frac{\tau(q)}{P(q)}\right). \qquad (3.42)$$

Substituting for $\phi(q)$ in equation 3.32 produces,

$$a_0 = \int_0^{\infty} P(q)\left(\left(\frac{(A\alpha^{\frac{1}{2}} - \lambda a_0^{\frac{1}{2}})}{A\alpha^{\frac{3}{2}}a}\right)\left(\frac{\tau(q)}{P(q)}\right)\right)^2 dq. \qquad (3.43)$$

Rearranging equation 3.43 and using equation 3.14 I generate,

$$a_0 = \frac{(A\alpha^{\frac{1}{2}} - \lambda a_0^{\frac{1}{2}})^2}{A^2\alpha^3 a}. \qquad (3.44)$$

Manipulating equation 3.44 produces,

$$(\lambda^2 - A^2\alpha^3 a)a_0 - 2A\lambda\alpha^{\frac{1}{2}}a_0^{\frac{1}{2}} = -A^2\alpha. \qquad (3.45)$$

Equation 3.45 is a quadratic equation in $a_0^{\frac{1}{2}}$,the solutions of which are given below.

$$a_0^{\frac{1}{2}} = \frac{A\alpha^{\frac{1}{2}}(\lambda \pm A\alpha^{\frac{3}{2}}a^{\frac{1}{2}})}{(\lambda^2 - A^2\alpha^3 a)} \qquad (3.46)$$

On rearranging equation 3.46 the result is,

$$a_0^{\frac{1}{2}} = \frac{A\alpha^{\frac{1}{2}}}{(\lambda \pm A\alpha^{\frac{3}{2}}a^{\frac{1}{2}})}. \qquad (3.47)$$

Substituting for $a_0^{\frac{1}{2}}$ in equation 3.42 produces,

$$\phi(q) = \left(\frac{1}{A\alpha^{\frac{3}{2}}a}\right)\left(A\alpha^{\frac{1}{2}} - \lambda\left(\frac{A\alpha^{\frac{1}{2}}}{(\lambda \pm A\alpha^{\frac{3}{2}}a^{\frac{1}{2}})}\right)\right)\left(\frac{\tau(q)}{P(q)}\right). \tag{3.48}$$

After simplifying equation 3.48 reduces to.

$$\phi(q) = \left(\frac{1}{\alpha a\left(1 \pm \frac{\lambda}{A\alpha^{\frac{3}{2}}a^{\frac{1}{2}}}\right)}\right)\left(\frac{\tau(q)}{P(q)}\right) \tag{3.49}$$

It is useful to define a new variable ω.

$$\omega = \frac{1}{\left(1 \pm \frac{\lambda}{A\alpha^{\frac{3}{2}}a^{\frac{1}{2}}}\right)} \tag{3.50}$$

Rearranging equation 3.50 to get λ, the result is.

$$\lambda = \pm\frac{(1-\omega)A\alpha^{\frac{3}{2}}a^{\frac{1}{2}}}{\omega} \tag{3.51}$$

Substituting for λ in equation 3.49 generates,

$$\phi(q) = \left(\frac{\omega}{\alpha a}\right)\left(\frac{\tau(q)}{P(q)}\right). \tag{3.52}$$

Substituting for $\phi(q)$ in equation 3.11 I obtain,

$$\int_0^\infty q^{n-1}\tau(q)\left(\left(\frac{\omega}{\alpha a}\right)\left(\frac{\tau(q)}{P(q)}\right)\right)dq = \frac{1}{\alpha}. \tag{3.53}$$

Solving equation 3.53 for ω, and using equation 3.14 the result is.

$$\omega = 1 \tag{3.54}$$

Substituting for ω in equation 3.51 I reach the result.

$$\lambda = 0 \tag{3.55}$$

Substituting for ω in equation 3.52 generates,

$$\phi(q) = \left(\frac{1}{\alpha a}\right)\left(\frac{\tau(q)}{P(q)}\right). \tag{3.56}$$

This completes my alternative derivation of the Vio kernel for filtering, using the calculus of variations with Lagrangian undetermined multipliers.

I shall now briefly outline the SCUBA matched filter technique outlined in the Serjeant et al. 2003 paper. The SCUBA filter assumes a white noise background but an RMS that varies from position to position. This variation is uncorrelated with the signal. In the Serjeant et al. 2003 paper, P is the PSF, the point spread function of the detector; N is the image noise; S is the image signal; W the weighted noise is given by equation 3.1; F is the best fit flux; ΔF is the error of the best fit flux. All of these quantities are functions of sky position and time. The symbol $\otimes$ represents convolution.The best fit flux was obtained by minimizing the χ^2, with respect to F. *(i,j)*, is the coordinates of a point in the image.

$$P = P(x,y) \tag{3.57}$$

$$N = N(i,j) \tag{3.58}$$

$$S = S(i,j) \tag{3.59}$$

$$F = F(i,j) \tag{3.60}$$

$\chi^2(i,j)$, is given by,

$$\chi^2(i,j) = \sum_{x,y}\left(\frac{S(i-x,j-y) - FP(x,y)}{N(i-x,j-y)}\right)^2. \tag{3.61}$$

In order to minimize the χ^2 with respect to F, we derive,

$$\frac{d\chi^2}{dF} = -2\left(\sum_{x,y}\left(\frac{(S(i-x,j-y) - FP(x,y))}{N(i-x,j-y)^2}\right)P(x,y)\right). \tag{3.62}$$

Putting the above equation to zero and solving for F, we get,

$$F(i,j) = \frac{\sum_{x,y} S(i-x,j-y)W(i-x,j-y)P(x,y)}{\sum_{x,y} W(i-x,j-y)P(x,y)^2}. \tag{3.63}$$

This last equation can be written in the form,

$$F = \frac{(SW) \otimes P}{W \otimes P^2}. \tag{3.64}$$

I will now outline the procedure known as propagating errors in F, to obtain ΔF. ΔS is the error in the image signal. ΔS is given by,

$$\Delta S(i,j) = N(i,j). \tag{3.65}$$

From 3.1 and rearranging we obtain,

$$N(i,j) = \frac{1}{(W(i,j))^{\frac{1}{2}}}. \tag{3.66}$$

Hence the error of the best fit flux is given by,

$$\Delta F = \left(\sum_{x,y}\left((\frac{\Delta S(i-x,j-y))^2(W(i-x,j-y))^2(P(x,y))^2}{(\sum_{x,y} W(i-x,j-y)(P(x,y))^2)^2}\right)\right)^{\frac{1}{2}}. \tag{3.67}$$

Substitute for *N(i,j)* in equation 3.65 to get,

$$\Delta S = \frac{1}{(W(i,j))^{\frac{1}{2}}}. \tag{3.68}$$

Substitute for ΔS in equation 3.67 and simplifying we obtain,

$$\Delta F = \left(\frac{1}{\sum_{x,y} W(i-x,j-y)(P(x,y))^2}\right)^{\frac{1}{2}}. \tag{3.69}$$

The previous equation can be written in the form,

$$(\Delta F)^2 = \frac{1}{W \otimes P^2}. \tag{3.70}$$

Further simplifying equation 3.70 get the result,

$$\Delta F = \frac{1}{\sqrt{W \otimes P^2}}. \tag{3.71}$$

The signal to noise map is given by $F/\Delta F$.

$$\frac{F}{\Delta F} = \frac{(SW) \otimes P}{\sqrt{W \otimes P^2}} \tag{3.72}$$

This completes my summary of the SCUBA matched filter outlined in the Serjeant et al. 2003 paper.

3.2.2 The typos in the Vio et al paper

In the Vio et al. 2002 paper there is a typographical error, the expression for the variance of $w(\overrightarrow{b}; \phi)$ is given in the paper as,

$$\sigma^2(\phi) = \alpha \int_0^\infty q^{n-1} P(q) \phi(q) dq. \tag{3.73}$$

The correct expression for the variance of $w(\overrightarrow{b}; \phi)$ is given by,

$$\sigma^2(\phi) = \alpha \int_0^\infty q^{n-1} P(q) \phi^2(q) dq. \tag{3.74}$$

This is clearly a typographical error, because later sections in the Vio et al. 2002 paper quote the correct result. I will now briefly outline how this correct result is obtained. The variables are all defined in the previous section. I will now restate the definitions and explain the key results. The mean of $w(\overrightarrow{b}; \phi)$ is given below,

$$\mu(\overrightarrow{b}; \phi) = \alpha \int_0^\infty q^{n-1} s(q) \phi(q) e^{i(\overrightarrow{q} \cdot \overrightarrow{b})} dq. \tag{3.75}$$

The variance of $w(\overrightarrow{b}; \phi)$ is defined below,

$$\sigma^2(\phi) = E[w^2(\overrightarrow{b}; \phi)] - \mu^2(\overrightarrow{b}; \phi). \tag{3.76}$$

This can be in an equivalent form,

$$\sigma^2(\phi) = E[w(\overrightarrow{b};\phi)w^*(\overrightarrow{b};\phi)] - \mu(\overrightarrow{b};\phi)\mu^*(\overrightarrow{b};\phi). \tag{3.77}$$

Substituting in equation 3.77 for w and μ, using integration variable q for w, and μ, and using variable, $\acute{q}$, for w^* and μ^*, you get,

$$\sigma^2(\phi) = I_1 - I_2 \tag{3.78}$$

$$I_1 = E[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y(\overrightarrow{q})y^*(\overrightarrow{q'})\phi(q)\phi^*(\acute{q})e^{i((\overrightarrow{q}-\overrightarrow{q'}).\overrightarrow{b})}d\overrightarrow{q}d\overrightarrow{q'}] \tag{3.79}$$

$$I_2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\overrightarrow{q})s^*(\overrightarrow{q'})\phi(q)\phi^*(q')e^{i((\overrightarrow{q}-\overrightarrow{q'}).\overrightarrow{b})}d\overrightarrow{q}d\overrightarrow{q'}. \tag{3.80}$$

We need the following equations.

$$E[z(\overrightarrow{q})] = 0 \tag{3.81}$$

$$E[z^*(\overrightarrow{q'})] = 0 \tag{3.82}$$

$$y(\overrightarrow{q}) = \sum_{j=1}^{N} s_j(\overrightarrow{q}) + z(\overrightarrow{q}) \tag{3.83}$$

$$y^*(\overrightarrow{q'}) = \sum_{j=1}^{N} s_j^*(\overrightarrow{q'}) + z^*(\overrightarrow{q'}) \tag{3.84}$$

$$E[\sum_{j=1}^{N} s_j(\overrightarrow{q})] = s(\overrightarrow{q}) \tag{3.85}$$

$$E[\sum_{j=1}^{N} s_j^*(\overrightarrow{q'})] = s^*(\overrightarrow{q'}) \tag{3.86}$$

Substituting for $y(\overrightarrow{q})$ and $y^*(\overrightarrow{q'})$, in equation 3.78 and simplifying we get,

$$\sigma^2(\phi) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(T_{10} + T_{11} + T_{12} + T_{13})\phi(q)\phi^*(\acute{q})e^{i((\overrightarrow{q}-\overrightarrow{q'}).\overrightarrow{b})}d\overrightarrow{q}d\overrightarrow{q'} - I_3 \tag{3.87}$$

$$T_{10} = (E[\sum_{j=1}^{N} s_j(\overrightarrow{q})])(E[\sum_{j=1}^{N} s_j^*(\overrightarrow{q'})]) \tag{3.88}$$

$$T_{11} = (E[z^*(\overrightarrow{q'})])(E[\sum_{j=1}^{N} s_j(\overrightarrow{q})]) \tag{3.89}$$

$$T_{12} = (E[z_j(\overrightarrow{q})])(E[\sum_{j=1}^{N} s_j^*(\overrightarrow{q'})]) \tag{3.90}$$

$$T_{13} = E[z(\overrightarrow{q})z^*(\overrightarrow{q'})] \tag{3.91}$$

$$I_3 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(\overrightarrow{q})s^*(\overrightarrow{q'})\phi(q)\phi^*(\acute{q})e^{i((\overrightarrow{q}-\overrightarrow{q'}).\overrightarrow{b})}d\overrightarrow{q}d\overrightarrow{q'}. \tag{3.92}$$

From equations 3.81, 3.82, 3.85, 3.86 and 3.87 and simplifying we obtain,

$$\sigma^2(\phi) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} P(\overrightarrow{q})\delta^n(\overrightarrow{q} - \overrightarrow{q'})\phi(q)\phi^*(\acute{q})e^{i((\overrightarrow{q}-\overrightarrow{q'}).\overrightarrow{b})}d\overrightarrow{q}d\overrightarrow{q'}. \tag{3.93}$$

After further simplification we arrive at,

$$\sigma^2(\phi) = \int_{-\infty}^{\infty} P(\overrightarrow{q})\phi(q)\phi^*(q)d\overrightarrow{q}. \tag{3.94}$$

Next we obtain,

$$\sigma^2(\phi) = \int_{-\infty}^{\infty} P(\overrightarrow{q})\phi^2(q)d\overrightarrow{q}. \tag{3.95}$$

Finally transforming to n dimensional polar coordinates and the doing the integrations that can be done straight away, we come up with,

$$\sigma^2(\phi) = \alpha\int_{0}^{\infty} q^{n-1}P(q)\phi^2(q)dq. \tag{3.96}$$

.

3.2.3 The modified kernel for filtering

S(t) is the signal as a function of time. $\tilde{S}(k)$ is the Fourier transform of *S(t)*, and k is the Fourier space variable. FT is the Fourier transform operator. FT^{-1} is the inverse Fourier transform operator.

$$\tilde{S}(k) = FT(S(t)) = a(\int_{-\infty}^{\infty} S(t)e^{-ikt}dt) \tag{3.97}$$

where a is a constant depending on the definition of the Fourier transform being used. There are a number of different definitions used in practice. For example the definition of the Fourier transform in the Vio et al. 2002 paper is different from the definition used in the computer programming language IDL version 8.0. *C(t)* is the point spread function, PSF. The PSF is the shape of the image of a point source in the observing instrument. $\tilde{C(k)}$ is the Fourier transform of *C(t)*.

$$\tilde{C(t)} = FT(PSF) \tag{3.98}$$

V(t) is the Vio, kernel.

$$V(t) = AFT^{-1}\left[\frac{\tilde{C(k)}}{P(k)}\right] \tag{3.99}$$

where *P(k)* is the power spectrum of the noise and A is constant of proportionality. In practice this means we take the input signal as an approximation of the noise. Hence *P(k)* is given by

$$P(k) = |\tilde{S(k)}|^2. \tag{3.100}$$

From equation 3.99 we get

$$\tilde{V(k)} = B\frac{\tilde{C(k)}}{P(k)}. \tag{3.101}$$

Where B is another constant of proportionality. Substituting for *P(k)* in equation 3.101 we find that

$$\tilde{V(k)} = B\frac{\tilde{C(t)}}{|\tilde{S(k)}|^2}. \tag{3.102}$$

$P(S \otimes V)$ is the power spectrum of $S \otimes V$, and $\otimes$ is the convolution operator

$$P(S \bigotimes V) = B|FT(S \bigotimes V)|^2. \tag{3.103}$$

Using the properties of Fourier transforms and convolutions, it follows that,

$$P(S \bigotimes V) = B|\tilde{S(k)}\tilde{V(k)}|^2. \tag{3.104}$$

Substituting for $V\tilde{(}k)$ in equation 3.104 and simplifying you obtain

$$P(S \bigotimes V) = B\frac{|C\tilde{(}k)|^2}{|S\tilde{(}k)|^2}. \tag{3.105}$$

This is not a white noise spectrum. Therefore, the Vio approach leads to a reddening of the filtered timeline. The Vio paper obtains the kernel which gives the optimum signal to noise. However, it increases the likelihood of bogus source detections. I will now outline the alternative approach to obtain a kernel for timeline filtering. Instead of optimizing the signal to noise, we deduce a kernel that whitens the power spectrum of the filtered timeline. $V_S(t)$ is the modified kernel.

The Fourier transform of $V_S(t)$ is $V_S\tilde{(}k)$.

$$V_S(t) = FT^{-1}\left(\frac{C\tilde{(}k)}{\sqrt{P(k)}}\right) \tag{3.106}$$

From equation 3.118 we get,

$$V_S\tilde{(}k) = \frac{C\tilde{(}k)}{\sqrt{P(k)}}. \tag{3.107}$$

Substituting for *P(k)* in equation 3.128 and simplifying the resulting equation yields

$$V_S\tilde{(}k) = \frac{C\tilde{(}k)}{|S\tilde{(}k)|}. \tag{3.108}$$

$P(S \otimes V_S)$ is the power spectrum of $S \otimes V_S$.

$$P(S \bigotimes V_S) = |FT(S \bigotimes V_S)|^2 \tag{3.109}$$

Applying the properties of Fourier transforms and convolutions the next equation follows,

$$P(S \bigotimes V_S) = |S\tilde{(}k)V_S\tilde{(}k)|^2. \tag{3.110}$$

Substituting for $V_s\tilde{(}k)$ in equation 3.110 and simplifying the result is,

$$P(S \bigotimes V_S) = |C\tilde{(}k)|^2. \tag{3.111}$$

3.2.4 The SNR penalty for white noise

Since we are now using the modified kernel as opposed to the Vio kernel, to do the timeline filtering, the signal to noise, SNR, will be less than optimal. I will now derive an analytic expression for the ratio of the signal to noise obtained using the modified kernel to the signal to noise obtained using the Vio kernel. Later I will give the numerical value for this ratio for each PACS pixel for the various data sets.

We start with the following relations:

$$\alpha = \frac{2\pi^{\frac{n}{2}}}{\Gamma\left(\frac{n}{2}\right)} \tag{3.112}$$

$$a = \int_0^\infty \frac{q^{n-1}\tau^2(q)dq}{P(q)} \tag{3.113}$$

$$\int_0^\infty q^{n-1}\tau(q)\phi(q)dq = \frac{1}{\alpha} \tag{3.114}$$

$$\phi(q) = \left(\frac{1}{\alpha a}\right)\left(\frac{\tau(q)}{P(q)}\right) \tag{3.115}$$

$$D(\phi) = A\alpha^{\frac{1}{2}}a^{\frac{1}{2}} \tag{3.116}$$

$$D(\phi) = \frac{A\alpha^{\frac{1}{2}}\left(\int_0^\infty q^{n-1}\tau(q)\phi(q)dq\right)}{\left(\int_0^\infty q^{n-1}P(q)\phi^2(q)dq\right)^{\frac{1}{2}}}. \tag{3.117}$$

α is the conversion factor that is required to convert the integrals to n dimensional spherical polar coordinates. a is the value of an integral that occurs in many of the analytic expressions. ϕ is the Vio kernel. q is the Fourier space variable. τ is the PSF. P is the power spectrum of the noise. A is the flux of the source. S is the expectation value of the signal. $D(\phi)$ is the signal to noise obtained when using the Vio kernel. ϕ_S is the modified kernel. Ω is the conversion factor that ensures that the modified kernel satisfies the constraint $\mu(\vec{O}; \phi_S) = A$. $D(\phi_S)$ is the signal to noise obtained when using the modified kernel. $\mu(\vec{O}, \phi_S)$ is the mean of the signal obtained when using the modified kernel. $\sigma^2(\phi_S)$ is the variance of the signal obtained when using the modified kernel. b is the value of a second integral that occurs in many of the analytic expressions. c is the value of a third integral that also occurs in many of the analytic expressions.

Equation 3.116 is obtained by substituting for $\phi(q)$ in equation 3.117, using equation 3.113 and simplifying.

$$\phi_S(q) = \frac{\Omega\tau(q)}{\sqrt{P(q)}} \tag{3.118}$$

$$D(\phi_S) = \frac{\mu(\vec{O};\phi_s)}{\sigma(\phi_S)} \tag{3.119}$$

$$\int_0^\infty q^{n-1}\tau(q)\phi_S(q)dq = \frac{1}{\alpha} \tag{3.120}$$

$$\mu(\vec{O};\phi_S) = \alpha\left(\int_0^\infty q^{n-1}S(q)\phi_S(q)dq\right) \tag{3.121}$$

$$\sigma^2(\phi_S) = \alpha\left(\int_0^\infty P(q)\phi_S^2(q)dq\right) \tag{3.122}$$

$$b = \int_0^\infty \frac{q^{n-1}\tau^2(q)dq}{\sqrt{P(q)}} \tag{3.123}$$

Substituting for $\phi_S(q)$ in equation 3.120, using equation 3.123, we obtain,

$$\Omega b = \frac{1}{\alpha}. \tag{3.124}$$

Solving equation 3.124 for Ω we obtain,

$$\Omega = \frac{1}{\alpha b}. \tag{3.125}$$

$$S(q) = A\tau(q) \tag{3.126}$$

Substituting for *S(q)* in equation 3.121 and simplifying the result is,

$$\mu(\vec{O};\phi_S) = A\alpha\left(\int_0^\infty q^{n-1}\tau(q)\phi_s(q)dq\right). \tag{3.127}$$

Substituting for Ω in equation 3.118 and simplifying we get,

$$\phi_S(q) = \left(\frac{1}{ab}\right)\left(\frac{\tau(q)}{\sqrt{P(q)}}\right). \tag{3.128}$$

Substituting for $\phi_S(q)$ in equation 3.127, using equation 3.123 and simplifying we get,

$$\mu(\overrightarrow{O};\phi_s) = A. \tag{3.129}$$

$$c = \int_0^\infty q^{n-1}\tau^2(q)dq \tag{3.130}$$

Substituting for $\phi_S(q)$ in equation 3.122, using equation 3.130 and simplifying we obtain,

$$\sigma(\phi_S) = \frac{c^{\frac{1}{2}}}{a^{\frac{1}{2}}b}. \tag{3.131}$$

Substituting for $\mu(\overrightarrow{O};\phi_S)$ and $\sigma(\phi_S)$, in equation 3.119 and simplifying we get,

$$D(\phi_S) = \frac{Aa^{\frac{1}{2}}b}{c^{\frac{1}{2}}}. \tag{3.132}$$

r is the ratio of the signal to noise obtained using the modified kernel to the signal to noise obtained using the Vio kernel, an expression for which is given below. It is arrived at by dividing $D(\phi_S)$ by $D(\phi)$, and simplifying.

$$r = \frac{D(\phi_S)}{D(\phi)} = \frac{b}{a^{\frac{1}{2}}c^{\frac{1}{2}}} \tag{3.133}$$

I will now outline the key steps in obtaining numerical estimates of the ratio of signal to noise obtained using the modified kernel to the signal to noise obtained using the Vio kernel. Firstly I will substitute for a, b and c in equation 3.133 to obtain

$$r = \frac{D(\phi_S)}{D(\phi)} = \frac{\int_0^\infty \frac{q^{n-1}\tau^2(q)dq}{\sqrt{P(q)}}}{\left(\int_0^\infty \frac{q^{n-1}\tau^2(q)dq}{P(q)}\right)^{\frac{1}{2}}\left(\int_0^\infty q^{n-1}\tau^2(q)dq\right)^{\frac{1}{2}}}. \tag{3.134}$$

For timeline filtering $n = 1$. Substituting for n in equation 3.134 and simplifying yields,

$$r = \frac{D(\phi_S)}{D(\phi)} = \frac{\int_0^\infty \frac{\tau^2(q)dq}{\sqrt{P(q)}}}{\left(\left(\int_0^\infty \frac{\tau^2(q)dq}{P(q)}\right)\left(\int_0^\infty \tau^2(q)dq\right)\right)^{\frac{1}{2}}}. \tag{3.135}$$

Writing the above expression in array notation for, IDL.

$$r = \frac{D(\phi_S)}{D(\phi)} = \frac{\sum_{i=1}^{N}\left(\frac{\tau^2(q_i)}{\sqrt{P(q_i)}}\right)}{\left(\left(\sum_{i=1}^{N}\left(\frac{\tau^2(q_i)}{P(q_i)}\right)\right)\left(\sum_{i=1}^{N}\tau^2(q_i)\right)\right)^{\frac{1}{2}}} \tag{3.136}$$

r is the symbol used for the ratio of the signal to noise obtained using the modified kernel to the signal to noise obtained using the Vio kernel.

$$r = \frac{D(\phi_S)}{D(\phi)} \tag{3.137}$$

The following variables are used in my timeline filtering programme.

$$a = \sum_{i=1}^{N}\left(\frac{\tau^2(q_i)}{P(q_i)}\right) \tag{3.138}$$

$$b = \sum_{i=1}^{N}\left(\frac{\tau^2(q_i)}{\sqrt{P(q_i)}}\right) \tag{3.139}$$

$$c = \sum_{i=1}^{N}(\tau^2(q_i)) \tag{3.140}$$

$$r = \frac{D(\phi_S)}{D(\phi)} = \frac{b}{(ac)^{\frac{1}{2}}} \tag{3.141}$$

I applied my filtering routines to the H-ATLAS Science Demonstration Phase (SDP) data. The SDP data contains four data sets, red70, red71, green70, and green71. Red in the name specifies a wavelength of 160μm and an array of PACS pixels of thirty two by sixteen or five hundred and twelve in total. Green in the name specifies a wavelength of 100μm and an array of PACS pixels of sixty four by thirty two or two thousand and forty eight in total. The 70 or 71 in the name specifies the observation.

Quantity	Value
minimum	0.608
maximum	0.703
mean	0.654
median	0.653
standard dev	0.00923
skewness	2.03
kurtosis	10.6

Table 3.1: The summary of the statistics of the ratio data of the red70 data set.

71 scans in a direction perpendicular to that of 70. They also start scanning from different locations in the field.

Table 3.1 shows the statistics of the red70 data set.

Table 3.2 shows the statistics of the red71 data set.

Table 3.3 shows the statistics of the green70 data set.

One of the elements is a NAN, standing for not a number. It probably means division by zero or a very small number has occurred. In this case I give the statistics of the ratio for all values except the one that is a, NAN. Table 3.4 shows the statistics of the green71 data set.

The overall minimum value of the ratio for the data sets, red70, red71, green70 and green71, is, 0.608. The overall maximum of ratio for the data sets, red70, red71, green70 and green71, is, 0.703. The overall range of ratio for the data sets, red70, red71, green70 and green71, is, 0.0953. The overall range is obtained by subtracting the overall minimum value from the overall maximum value. Hence for the data sets,

Quantity	Value
minimum	0.608
maximum	0.703
mean	0.657
median	0.657
standard dev	0.00841
skewness	2.04
kurtosis	13.5

Table 3.2: The summary of the statistics of ratio data for the red71 data set.

Quantity	Value
minimum	0.640
maximum	0.702
mean	0.672
median	0.672
standard dev	0.00602
skewness	1.21
kurtosis	6.36

Table 3.3: The summary of the statistics of the ratio data for the green70 data set.

Quantity	Value
minimum	0.643
maximum	0.702
mean	0.678
median	0.678
standard dev	0.00484
skewness	0.375
kurtosis	7.13

Table 3.4: The summary of the statistics of the ratio data for the green71 data set.

red70, red71, green70 and green71, the ratio value lies roughly in the range from 61 percent to 70 percent. Therefore filtering the timelines to produce a white noise spectrum reduces the signal-to-noise to 61-70% of the maximum possible.

3.3 The PACS pipeline

3.3.1 The PACS data reduction methodology

3.3.1.1 The first version of the PACS pipeline programme

The PACS instrument operates on three wavelengths, only two of which can be used at the same time. These wavelengths are 70μm, defined to be blue, 100μm, defined to be green, and 160μm, defined to be red. The pair I am using are the green and the red.

The data from PACS comes to us as a data cube, that gives us the PACS flux for a given wave band, for a given pixel of the detector as a function of time. The flux is in Janskys per pixel. The input timelines of the PACS data have been calibrated and deglitched in Herschel Interactive Processing Environment HIPE. The first dimension of the data cube specifies the time of the observation. The second and third dimensions of the data cube specify the PACS pixel of the detector. Other data comes to us from the PACS instrument, such as the right ascension α and declination δ of a given PACS pixel as a function of time. We also get data which allows us to deglitch the data, that is correct for calibration spikes and etcetera.

The PACS detector scans across a specified region of the sky using different starting points, scan directions and scan rates depending on the run options specified. It automatically scans the area of the sky under observation in two different directions that are at right angles to each other. The red70 and red71 data sets are essentially the same area of the sky observed at the same time scanned in perpendicular directions. The same is true of the green70 and green71 data sets. A data set's name is in two parts: the colour red or green for the wave band and the number 70 or 71 that specifies the observation. This allows us to detect some bogus signals such as cosmic rays. This is done by looking for signals that do not occur in the same place in the sky at the same time in both scan directions. If signals do occur at the same place in the sky at the same time, to within the limits of observational error, in both scan directions, then they are more likely to be genuine sources.

The red70 data set has a time span of 282068 time steps, the precise amount of time will depend on the scan rate. The red71 data set has a time span of 285826 time steps. The green70 data set has a time span of 141065 time steps. The green71 data set has a time span of 142918 time steps. The red70 data cube has a total size of 144418816 elements. The red71 data cube has a total size of 146342912 elements. The green70 data cube has a total size of 288901120 elements. The green71 data cube has a total size of 292696064 elements.

For a given run the PACS detector scans across a roughly rectangular region of the sky. The deviation from a perfect rectangle is due to lag and over-run of the PACS instrument. It starts at one of the four corners of the rectangle and starts to move towards one of the two adjacent corners of the rectangle. When it reaches the target corner of the rectangle it then moves along one step on the side of the rectangle perpendicular to its initial scan direction, then heads back in the opposite direction antiparallel to its original course. When it reaches the opposite side of the rectangle one pixel along on that side from its starting point, it then moves another step along this side of the rectangle. Then it heads back in the same manner as before repeating the scanning procedure until the whole rectangle has been scanned.

The first version of the PACS pipeline programme was written by my supervisor, Stephen Serjeant, in IDL version 7.1.Table 3.5 shows the list of the main steps of the first version of the PACS pipeline programme.

I will now outline briefly what happens in each of these fifteen sections. I restrict myself to the in house routines and exclude built in IDL routines.

In section one, the programme sets up the input and output directories by specifying the appropriate string variables. Then it sets up the file name extensions for the save files that are being used by defining the relevant string variables. Subsequently the read in file format is defined. The data we get from Herschel PACS HIPE pipeline contains calibration flashes. In the timeline data for a given PACS pixel they take the form of a narrow spike of given height. In order to eliminate them from the data

Step	Description	Notes
1	Looking for input directory and initializing	
2	Specify file names and directory	
3	Reading in the image data	
4	Specifying variable and header for sky projection	
5	Create noise array	
6	Flash identification and masking	
7	Time line filtering	
8	Calibration flash	
9	Sky subtraction	
10	Noise array	
11	Save data	
12	Projecting to sky	
13	Save data	
14	Two D filtering	
15	Saving final results	

Table 3.5: Description of the stages of the first version of the PACS pipeline programme.

we need to define a flash threshold. Any signal greater than this level is set to zero and the noise to a large value. This is done for that time index and PACS pixel. The flash threshold for calibration flashes is defined. The full width at half maximum, fwhm, for the 2D gaussian used for 2D filtering is specified. The pixel size is defined, which is used in making a header for projecting data on to the sky. It is also used in specifying the size of the arrays, signal and noise, that are projected on to the sky. A mask flashes option is defined, so the programme knows how to handle calibration flashes. The variable width is specified which is used in the section of the programme that does the sky subtraction.

In section two, the programme specifies the file names and the directory that they are saved to. If the saves directory does not exist then it is created.

In section three, the programme reads in the image data. Originally the PACS data came to us as a FITS file, which had to be split up into twenty, IDL save sets that can be read by IDL. Each of these files contain different information plus a header that outlines the contents of each file. There is a header for each of these files.

In section four, the programme uses the variable pixel size and the ra and dec arrays, to create the header for projecting the image and noise on to the sky.

In section five, the programme creates the noise array. The noise for each PACS pixel is deduced from the root mean squared (RMS) of the timeline data for that pixel as a function of time. If the noise value comes out to be zero it is then set to the value 1×10^{10}, so the noise array will have a value that is time independent. Hence it will have a value independent of the value of the first dimension of the noise array. It will have a value that is dependent on the values of the second and third dimensions of the array, so the noise will be pixel dependent only.

In section six, the programme eliminates calibration flashes by masking. When the timeline data is greater than the flash threshold, specified in section one of the programme, it changes the noise to the value 1×10^{10}.

In section seven, the programme uses the SCUBA matched filter technique, Serjeant et al. 2003, using a Gaussian kernel, to do the smoothing needed for beam switching. The array weight, used in section 3.1.1. of this thesis, is given by,

$$weight = \frac{1}{noise^2}. \tag{3.142}$$

When the noise is equal to zero, the weight is also set to zero for this pixel and time index. If the noise is greater than 1×10^5, then the weight is also set to zero for this pixel and time index. A normalization factor is needed, derived empirically, which is given by,

$$normalization = \frac{total(kernel^2)}{total(kernel)}. \tag{3.143}$$

In section eight, the programme initializes the array filtered noise array to the value 1×10^{10}, at the time indices near a calibration flash. It does this using the flash mask defined in section six of the programme.

In section nine, the programme does the sky subtraction using the variable width, which was defined in section one of this programme. A median boxcar sky subtraction is carried out on the array filtered timeline. When you get close to the edges of the PACS field you cannot obtain an accurate background measurement, as a result we set the sky_noise equal to 1×10^{10} at these locations. The filtering in section nine of the programme is only there to create a model of the $\frac{1}{f}$ fluctuations. The model is then subtracted so the timelines have not been filtered for point sources.

In section ten, the programme specifies a mask array which states where the noise in the filtered signal is greater than 1×10^5. This means that it should include all the calibration flash locations as well.

In section eleven, the programme saves all the timeline results. Each timeline array is saved to its own file inside the saves directory that was set up in section two of the programme.

In section twelve, the programme projects the filtered timeline data on to the sky, using a noise-weighted coadd. In order to find the flux at a given right ascension and declination it is necessary to use the astrolib subprogramme adxy to find the appropriate pixel location as the inputs for the skyimage array.

In section thirteen, the programme saves all the zerofootprint results. The term zerofootprint means data that has been timeline filtered and has been projected to the sky, but has not undergone any spatial filtering. Each zerofootprint array is saved to its own IDL save set inside the output directory. Each file has its own header and the array variables are two dimensional as they are sky projected results that have undergone timeline filtering but not 2D filtering.

In section fourteen, the programme produces a two dimensional Gaussian that is needed as an input for the two dimensional spatial filtering routine.

Next the programme does the two dimensional spatial filtering on the data projected onto the sky. This section of the programme uses the algorithm in equations 3.64 and 3.70.

In section fifteen, the programme saves all the filtered results. Each filtered array is saved to its own IDL save set file inside the output directory. Each file has its own header and the array variables are two dimensional as they are sky projected results that have undergone a subtraction of the $\frac{1}{f}$ drifts and two dimensional spatial filtering.

The first version of the PACS pipeline programme, takes the raw timeline data and smoothes it using the SCUBA matched filtering technique. Then the beam switching is done. Beam switching takes the smoothed timeline and obtains the sky model. If there is a source at location x and we take the two positions $x-\delta$ and $x+\delta$, where δ is a constant. Then it evaluates the mean of the smoothed timeline data at positions $x-\delta$ and $x+\delta$ and this is the value of the sky model at x. Then we subtract the sky model from the raw timeline to obtain the sky subtracted timeline. Next we project the data onto the sky, and then perform the two dimensional spatial filtering.

3.3.1.2 Calibration of the first PACS pipeline

I will now specify how the calibration factors for the first PACS pipeline programme, are deduced. There are two calibration factors, one for the green70 and green71 data sets, and one for the red70 and red71 data sets.

The green data comes in the form of 3D data cube in, Jy/PACS pixel. The green PACS pixel is 3.2" by 3.2" in angular size on the sky. This is put into a zerofootprint image with 2"pixels for the green image. Therefore to convert the zerofootprint image to flux per 2" pixel we need to multiply by $\left(\frac{2}{3.2}\right)^2$ for the green image. The total of the PSF is 14.6433 for the green image. The noise-weighted filtering routine reports that the best scaling of the PSF that is flux=1, when the image is exactly the same as the PSF.

Here are the steps needed to calibrate the green image.

1. Skyimage=Skyimage$\times\left(\frac{2}{3.2}\right)^2$, which converts to flux per 2" pixel.
2. Run the noise-weighted source extraction and output its flux.

3. Flux=Flux×14.6433. This converts the flux to Jy per beam.

4. Flux=$1 \times 10^3 \times$Flux. This converts the flux to mJy per beam.

This could be done in one go, that is multiply by $1 \times 10^3 \times \left(\frac{2}{3.2}\right)^2 \times 14.6433$. The overall calibration factor for the green image is 1 count in the final map is 5.72003Jy per beam, or 5720.03mJy per beam.

The red data comes in the form of 3D data cube in, Jy per PACS pixel. The red PACS pixel is 6.4" by 6.4" in angular size on the sky. This is put into a zerofootprint image with 3" pixels for the red image. Therefore to convert the zerofootprint image to flux per 3" pixel we need to multiply by $\left(\frac{3}{6.4}\right)^2$ for the red image. The total of the PSF is 16.6608 for the red image.

The steps needed to calibrate the red image and follow the same pattern as the green image:

1. Skyimage=Skyimage$\times \left(\frac{3}{6.4}\right)^2$, which converts to flux per 3" pixel.

2. Run the noise-weighted source extraction and output its flux.

3. Flux=Flux×16.6608. This converts the flux to Jy per beam.

4. Flux=$1 \times 10^3 \times$Flux. This converts the flux to mJy per beam.

This could be done in one go, that is multiply by $1 \times 10^3 \times \left(\frac{3}{6.4}\right)^2 \times 16.6608$. The overall calibration factor for the red image is 1 count in the final map is 3.66082Jy per beam, or 3660.82mJy per beam.

3.3.1.3 The second version of the PACS pipeline programme

The second version of the PACS pipeline programme was written by my supervisor's PhD student, Ros Hopwood, who is now a post doctoral student at Imperial College London, in IDL version 7.1. However, the overall frame work of the programme is the same as the original programme. In the second version of the PACS pipeline programme the two sections of the original programme dealing with calibration flashes have been combined into one. Table 3.6 shows the list of the main steps of the second version of the PACS pipeline programme.

The second version of the PACS pipeline programme introduces the following three main changes.

1). There is more modularization, that is moving operations into subroutines.

2). It replaces masked regions with random noise in case the user uses the whole data set (instead of the non-masked data) for statistics.

3). Calibration block residuals are now dealt with during the HIPE de-glitching, and the mask has these positions flagged.

3.3.1.4 The third version of the PACS pipeline programme

Originally we did the timeline filtering with a Gaussian kernel and the SCUBA filtering, with beam switching to model the $\frac{1}{f}$ noise, which is then subtracted out, plus two dimensional spatial filtering with a Gaussian kernel and the SCUBA filtering. Now we do the timeline filtering with a modified kernel, for $\frac{1}{f}$ noise removal, using the noise weighted matched filtering, plus a cross-scan spatial filtering with a Gaussian kernel and the SCUBA filtering. Both methods use all the information in the image, though not necessarily optimally. The first method uses timeline and spatial filtering separately. The second method uses the timeline and spatial filtering together. The timeline filtering uses all the information in each row. The cross-scan filtering uses all the information in each column. In two dimensional spatial filtering the input image is filtered in both rows and columns. The new timeline filtering does timeline filtering

Step	Description	Notes
1	Looking for input directory and initializing	
2	Specify file names and directory	
3	Reading in the image data	
4	Specifying variable and header for sky projection	
5	Create noise array	
6	Calibration flash identification and removal	
7	Timeline filtering	
8	Sky subtraction	
9	Noise array	
10	Save data	
11	Project to sky	
12	Save data	
13	Two D filtering	
14	Save final results	

Table 3.6: Description of the stages of the second version of the PACS pipeline programme.

plus one dimensional spatial filtering in each row. Hence only cross-scan filtering is needed in each column.

In the second method the timeline filtering with the modified kernel and SCUBA noise-weighting does the timeline filtering used in the first method plus the one dimensional spatial filtering in the scan direction. In the second method the cross-scan filtering is done with a narrow Gaussian kernel and the SCUBA noise-weighted filtering. The Gaussian kernel used to do the cross-scan filtering in the second consists of three rows and a number of columns specified in the third version of the PACS pipeline programme. The first and third rows of the Gaussian kernel used to do the cross-scan filtering are zero everywhere. The second is a Gaussian function of column pixel with a full width at half maximum specified in the third version of the PACS pipeline programme. The reason the kernel is two dimensional for the cross-scan filtering is because it is not clear what IDL would do with a one dimensional kernel to convolve with a two dimensional image.

The version three of the PACS pipeline programme was written by myself in IDL version 7.1. I modified the second version of the PACS pipeline programme to use the modified timeline filtering and the cross-scan filtering. Modified timeline filtering is timeline filtering using the modified kernel set out section 3.1.3., using the noise weighted matched filtering. Cross-scan filtering is briefly outlined in the previous paragraph. This meant that two dimensional spatial filtering and sky subtraction are both no longer necessary. But, the main layout of the third version of the PACS pipeline programme is the same as the version two of the PACS pipeline programme. Table 3.7 shows the list of the main steps of the third version of the PACS pipeline programme.

Step	Description	Notes
1	Looking for input directory and initializing	
2	Specify file names and directory	
3	Reading in the image data	
4	Create noise array	
5	Calibration flash identification and removal	
6	Timeline filtering	
7	Noise array	
8	Save data	
9	Project to sky	
10	Filter in cross-scan direction	
11	Save data	
12	Scale deltaflux	
13	Save final results	

Table 3.7: Description of the stages of the third version of the PACS pipeline programme.

Section one of the third version of the PACS pipeline programme is the same as section one of the second version of the PACS pipeline programme.

Section two of the third version of the PACS pipeline programme is the same as section two of second version of the PACS pipeline programme.

In section three, the arrays, ra, dec and flux arrays are read in.

Section four of the third version of the PACS pipeline programme is the same as section five of the second version of the PACS pipeline programme.

Section five of the third version of the PACS pipeline programme is the same as section six of the second version of the PACS pipeline programme.

In section six, the kernel is made for each PACS pixel and the timeline filtering is done for each PACS pixel.

In section seven, the locations where the filtered noise is less than 1×10^{-10} are found. Then at these locations the filtered noise values are set to the value 1×10^{10}. This is necessary for when the image and noise data are projected onto the sky.

In section eight, the filtered data are saved in IDL save set.

Section nine, generates a header and a blank array for projecting the image and noise data on to the sky.

The programme then projects the image and noise timeline data on to the sky.

In section ten, the filtering in the cross-scan direction is done.

A Gaussian kernel for the cross-scan filtering is generated. Then the noise-weighted filtering is used to do the cross-scan filtering.

In section eleven, the projected un-filtered image and noise are saved disc files. The save files also have their appropriate headers. The signal-to-noise image is calculated. If the noise is zero the signal to noise is also set to zero. This array is also saved along with its header.

Then in a second group of saving data to files, the projected filtered image and noise along with their headers are saved.

In section twelve, a projected filtered signal-to-noise image is created, again setting it to zero where the noise is zero. Then this SNR is saved along with its header to a disc file. Then the noise is scaled to enforce an SNR with unit variance. The noise and snr are modified if the snr does not have unit variance, presumably because there is still some $\frac{1}{f}$ noise there, so you opt to rescale the noise values. This updated noise image is then saved.

In section thirteen, this SNR array along with its header are saved to disc.

In the PACS pipeline programme the third version, the subprogrammes I have written with my supervisor's assistance are: the Vio filtering, the definition of the master sky image for coadding, the timeline-to-sky coadding and the kernel generation. I had to make some changes to the main programme that constitutes the third version of the PACS pipeline programme. A listing of the third version of the PACS pipeline programme is given in the appendix.

3.3.1.5 The calibration of the third version of the PACS pipeline

The green PACS source catalogue was generated from the coadded green70 into green71 SNR data obtained from the output data produced by the third version of the PACS pipeline programme. The red PACS source catalogue was generated from the coadded red70 into red71 SNR data obtained from the output data produced by the third version of the PACS pipeline programme. The details of this process are outlined in sections 4.1., 4.2., and 4.3.

Next the catalogue of the sources seen in both the PACS red and green source catalogues was obtained. Then the catalogue of sources seen in the PACS red and green catalogues plus the SPIRE catalogue was obtained. See section 4.3. for details.

The catalogue of sources seen in the PACS red and green catalogues plus the SPIRE catalogue, is used to generate a flux calibration of the PACS green data plus the PACS red data, by comparison with the H-ATLAS official data products. I wrote an IDL programme to do this.

The coadding process of 70 and 71 data is outlined in section 4.1.

All the sources in the catalogue of sources in common to both the PACS red and green catalogues have their great circle distances on the sky from each and every SPIRE source determined. Those sources closer than the minimum angular distance, equal to the resolution of the SPIRE 250 micron images on Herschel (since it is the greatest resolution used to find sources) are determined. These sources are taken as being the same source. The catalogue numbers of our source and their source that are matched is recorded.

This information is used to obtain the relevant PACS fluxes, ours and the SPIRE ones. Then graphs of our flux against their flux are plotted. The data is sorted in ascending order in terms of flux data. The intercept and gradient of each graph is obtained. The sorting is necessary for the built in IDL routine that does the linear fitting. The straight lines of best fit are also plotted on their respective graphs, as a final check.

Our fluxes are obtained from the right ascension and declination of the sources, using the adxy astrolib IDL routine, along with the appropriate header, to get the pixel locations of the source. This is used as the input to the flux array to get our

PACS fluxes for both red and green wavebands. This gives the PACS red and green fluxes for all the sources in our source catalogue. In order to obtain the PACS fluxes for both red and green wavebands corresponding to the matched SPIRE source the catalogue number of our source is needed. Their PACS red and green fluxes are obtained from the catalogue number of their source.

The gradient of the green flux straight line of best fit for the graph of our green flux against their green flux is the green calibration factor. The gradient of the red flux straight line of best fit for the graph of our red flux against their red flux is the red calibration factor. A_g is the intercept of the green flux straight line of best fit. B_g is the gradient of the green flux straight line of best fit. A_r is the intercept of the red flux straight line of best fit. B_r is the gradient of the red flux straight line of best fit.

$$A_g = -0.0986 \tag{3.144}$$

$$B_g = 24.5 \tag{3.145}$$

$$A_r = -0.0901 \tag{3.146}$$

$$B_r = 3.00 \tag{3.147}$$

3.3.2 Timeline filtering

The timeline filtering routine has fourteen sections. I will summarize them briefly plus outline how they work.

1. This section converts the fwhm in to the standard deviation theta of the PSF, using $\theta = \frac{FWHM}{2\sqrt{ln(4)}}$.

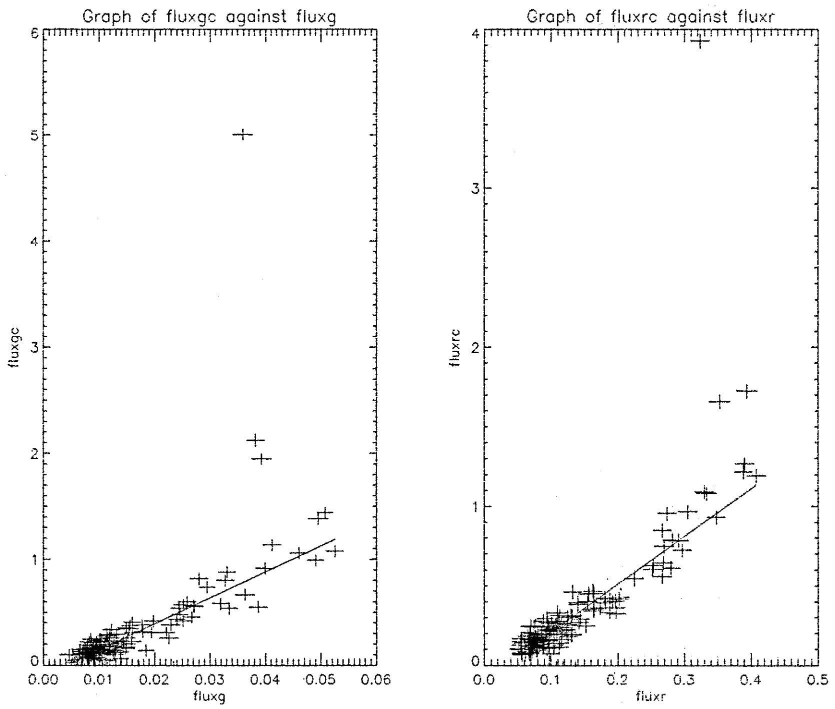


Figure 3.1: There are two graphs of our flux against their flux for both the green and red PACS wavebands.

2. This section computes the power frequency spectrum of the noise array.

3. This section finds the best fit values for d and gammal in $p = d * q^{-gamma1}$.

4. This section computes the PSF and kernel in Fourier space.

5. This section performs the inverse Fourier transform, to obtain the, kernel.

6. This section computes the input flux of the input signal plus fluctuations.

7. This section computes the signal to noise array.

8. This section adjusts the noise array to give the modified noise array, so that using the modified noise in evaluating the SNR array the SNR has variance one. This is done by finding the multiplier to produce the modified noise array.

9. This section computes the weighted noise array $W = \frac{1}{noise^2}$.

10. This section generates the array SW by multiplying element by element the input signals by the array weighted noise W. SW is defined as the signal muitiplied by the weight.

11. This section convolves SW with the kernel.

12. This section convolves W with the kernel squared.

13. This section computes the array output_timeline by dividing the result in section 11 by the result in section 12, see equation 3.64. The calculated PSF has the maximum value of 1 and total flux of total(psf). We need the output to be the total flux seen by the pixel as it scans across the source. In the absence of 1/f the kernel is just the PSF and a point source with a peak of 1.0 will come out with a reported peak flux of,

$$\frac{max(convol(truncated\_psf, truncated\_psf, /edge\_wrap))}{total(psf^2)}. \tag{3.148}$$

The PSF is truncated to save computation time; only those values that are approximately equal to zero are removed. This must be equal to 1.0 exactly regardless of the PSF. What we want it to report is the total flux in the PSF, so we need to multiply by total(psf). But in the case of 1/f noise, the kernel is different, so a point source with the shape of the PSF will have a reported flux of,

$$F = \frac{max(convol(truncated\_psf, kernel, /edge\_wrap))}{total(psf^2)}. \qquad (3.149)$$

We need to correct this so we need to divide by F and then multiply by the total(psf) for the reasons given above. This section also modifies the output noise array so that it has variance one after calibration.

14. This section computes the output flux of the output_timeline.

I will now elaborate on some of these steps.

This last part of section three of the timeline filtering routine does the smoothing of the power frequency spectrum in Fourier space. *P(k)* is the power frequency spectrum of the noise. In Vio et al. 2002, the power frequency spectrum of the noise was assumed to be fitted by a power law,

$$P(k) = Dk^{-\gamma} \qquad (3.150)$$

where D and γ are constants. However, the noise in our data is not well fitted by such a curve, so we had to use median smoothing to get the power frequency spectrum. From equation 3.150 you get,

$$\log_e P(k) = \log_e(Dk^{-\gamma}), \qquad (3.151)$$

after simplifying you obtain,

$$\log_e P(k) = \log_e D - \gamma \log_e k. \qquad (3.152)$$

I will make an exception for quoting IDL variables here. I will show the IDL code for this part of section three.

$$smoothing\_scale = 2000 \tag{3.153}$$

$$pfit = exp(median(alog(p\_average), smoothing\_scale)) \tag{3.154}$$

$$q\_for\_pfit = xn\_average{*}!dpi/max(xn\_average) \tag{3.155}$$

The small -q structure is not well reproduced by this smoothing, so an extrapolation from nearby values is used. This smoothed and extrapolated *P(k)* is used to generate the output kernel.

In section eight of the subprogramme we used the stdev_pclip astrolib routine in preference to astrolib routine sky_stats to obtain the standard deviation because of the digitization of the input data.

In section eleven of the subprogramme a convolution algorithm written by my supervisor is used. It takes twice the number of pixels in the kernel in the beginning of the SW array and reverses them and then convolves SW with the kernel. This deals with edge effects. Next is an in-house subprogramme written by my supervisor for removing long drifts. It is a boxcar median sky subtraction.

This subprogramme was necessary in the early pipeline versions. We retain it because the kernel is not perfect, perhaps because the kernel constructed from the power spectrum is not perfect. The median scale is the size of the entire kernel.

In section twelve, the array *w_x_p2* is generated by using the in house convolution function described in section eleven.

3.3.3 Cross-scan filtering

The cross-scan filtering section of the third version of the PACS pipeline programme consists of four subprogrammes. The first generates a blank sky image, a blank sky noise and a header needed for sky projecting, plus all of these outputs are inputs needed by the second subprogramme. The second does the sky projection using coadding with weighted noise. The coadding routine takes the PACS pixel and coadds it into the small sky pixel. This " coadding " is just a variance weighted mean average. If there was no data in the small sky pixel to begin with the small pixel is initialized to a very high noise value, about 1×10^{10}, and arbitrary signal for example zero, so the coadded value of the small sky pixel and PACS pixel is approximately equal to the PACS pixel on its own. The third subprogramme makes the kernel needed to do the cross-scan filtering. The fourth subprogramme, written by my supervisor, does the cross-scan filtering. See appendix A3 for a more detailed description of the cross-scan filtering section of the third version of the PACS pipeline programme.

Chapter 4

Results

4.1 Preview of chapter four

In chapter four, I will explain how the noise and SNR maps are produced. I will also outline how the SEDs and postage stamps of the PACS-only sources are obtained. Plus I will give their indentifications. Lastly I will produce the QSO stacks of my list of Quasar candidates.

4.2 Noise maps

The first step to constructing noise maps for the PACS red and green wavelengths, is to coadd the red70 data into the red71 data, and to coadd the green70 data into green71 data. The coadding algorithm is the same for both the PACS red and green wavelengths.

The coadding process just needs two output files from the third version of the PACS pipeline programme. The first file is the data file containing the PACS flux on the sky after both timeline and cross-scan filtering have been done. This file contains a two dimensional array of the PACS flux per sky pixel plus an appropriate header. The second file is the data file containing the PACS noise on the sky after both timeline and cross-scan filtering have been done.

I wrote two IDL programmes, one to coadd the red70 data into the red71 data, and the second to coadd the green70 data into to the green71 data. In both of these programmes the relevant PACS fluxes and noises are converted into the combined flux and combined noise, plus a new header is generated. In each case the output flux and noise from pipeline runs for 70 and 71 data sets is coadded for the red or green PACS data sets. The combined flux and combined noise two dimensional arrays and the new header are saved in appropriately named IDL save file. The combined flux f and combined noise n in terms of the input fluxes and noises are given below.

$$f = \frac{f_{71}/n_{71}^2 + f_{70}/n_{70}^2}{1/n_{71}^2 + 1/n_{70}^2} \tag{4.1}$$

$$ne = \frac{1}{\sqrt{1/n_{71}^2 + 1/n_{70}^2}} \tag{4.2}$$

f_{71} is the flux from the PACS pipeline run using 71 data set. n_{71} is the noise from the PACS pipeline run using the 71 data set. f_{70} is the flux from the PACS pipeline run using the 70 data set. n_{70} is the noise from the PACS pipeline run using the 70 data set. The routine also modifies arrays flux70 and noise70 to conform to the new header.

Since the arrays *flux70* and *flux71* contain no NaNs, and the arrays *noise70* and *noise71* contain no NaNs or zeros, it safe to use formulae for the combined flux arrays and combined noise arrays used in these programmes. It also means that the array combined flux contain no NaNs. In addition it also means that the array combined noise contains no NaNs or zeros.

4.3 SNR maps

In order to construct the signal to noise (SNR) maps for the PACS red and green wavelengths, I need to use the coadded red signal and the coadded red noise, to calculate the coadded red SNR, and similarly I need to utilize the coadded green signal and the coadded green noise, to evaluate the coadded green SNR. The determination of the SNR for both PACS red and green wavelengths is essentially the same just involves different data files.

The coadded signal and coadded noise correspond to the combined signal and combined noise two dimensional array outputs from the coadding programme. There will be two such output data sets corresponding to the red and green data sets. I have written two IDL programmes, one to generate the red coadded SNR, and the second to generate green coadded SNR. In both of these programmes the relevant PACS combined signal and combined noise is converted into the SNR and the same header is used from the input file that contained the combined signal s and combined noise n arrays. Then the SNR and the header are saved to an appropriately named IDL save file. The SNR is given by the equation below.

$$snr = \frac{s}{n} \tag{4.3}$$

Since the arrays combined signal and combined noise have no NaNs, and the array combined noise has no zeros, then array snr will have no NaNs.

4.4 SEDs of PACS-only sources and their IDs

4.4.1 The PACS-only sources for the first version of the PACS pipeline

In order to obtain PACS only sources, that is the sources seen in both the green and red PACS wavelengths but not seen in SPIRE, we need to construct the PACS green and red catalogues of sources. Initially we are using the output from the first version of the PACS pipeline programme, making use of the early science phase Herschel data.

The outputs for the four data sets, green70, green71, red70 and red71 for the first version of the PACS pipeline programme are obtained. Then the green70 data is coadded into the green71 data and the red70 is coadded into the red71 data. This procedure is outlined in section 4.1. Then coadded green signal and noise data are used to generate the coadded green signal to noise data. Similarly the coadded red signal and noise data are used to generate the coadded red signal to noise data. This procedure is laid out in section 4.2. For this data the coadding was originally done by my supervisor.

Two IDL programmes were written by myself to generate source catalogues for the PACS green and red data sets. The procedure is the same for both the green and red data sets, just different input and output files are used and made. The data file containing the appropriate signal to noise and header is restored. Then a connected pixel algorithm is used to construct the source catalogue. The inputs to this subroutine are the signal to noise array and the source detection threshold in standard deviations of the signal to noise, which is in this case six. The output sources are therefore known as 6σ sources.

The astrolib IDL routine xyad is used along with the header to get the right ascension and declination of the sources. The signal to noise of the sources is obtained from the pixel location of the sources. The right ascension and declination of each source is also saved to .input file which can be read by a person or by the Aladin programme.

Next the catalogue of sources in common to the PACS green catalogue and the PACS red catalogue is obtained by an IDL programme written by myself. The PACS green source catalogue data file is restored containing the right ascension, declination and signal to noise of each PACS green source. The PACS red source catalogue data file is restored containing the right ascension, declination and signal to noise of each PACS red source. Then the minimum distance on the sky between two sources from the source in the PACS green catalogue to a source in the PACS red catalogue for them to be considered the same source is defined. This minimum distance is equal to the resolution of the Herschel space observatory for the red PACS detector since it is the largest.

The Astrolib routine gcirc is used to find these distances between the red and green catalogue positions.

Then a catalogue of sources in common to the PACS green catalogue, PACS red catalogue and the SPIRE catalogue is obtained by an IDL routine which I wrote myself. The SPIRE catalogue of sources was obtained from the H-ATLAS consortium web site. This programme is very similar to the one used to generate the catalogue of sources in common to the PACS green catalogue and the PACS red catalogue.

The input files are the catalogue of sources in common to the PACS red and green catalogues and the SPIRE catalogue. The output files are the catalogues of sources in common to the PACS red and green catalogues and the SPIRE catalogue as an IDL save set plus .input file.

Next the catalogue of sources that are in both the PACS red and green catalogues but are not in the SPIRE catalogue is obtained. I did this by writing an IDL programme. The data file containing the catalogue of sources in common to both the PACS red and green catalogues is restored. The data file containing the catalogue of sources in common to both PACS red and green catalogues plus the SPIRE catalogue as well is restored. This file contains the SPIRE catalogue number, the red catalogue number and the green catalogue number for each source.

If a source is in common to both the PACS red and green catalogues and the SPIRE catalogue, then it is flagged as a triple match. The number in the red catalogue, the number in the green catalogue, the right ascension and the declination of all sources in the catalogue of sources in the PACS red and green catalogues but not in the SPIRE catalogue, are saved to an IDL save file. The right ascension and declination of these sources are also saved to an .input file.

There are eighty eight sources that are seen in both PACS red and green wavelengths but not seen in the SPIRE instrument. Then I wrote three IDL programmes to generate postage stamps for each of these sources for all the SPIRE wavelengths. The SPIRE wavelengths are 250μm, 350μm and 500μm. SPIRE is a more sensitive instrument than PACS, so sources seen in PACS but not seen in SPIRE are unexpected and interesting, or wrong!

The astrolib routine adxy is used to convert the right ascension and declination of

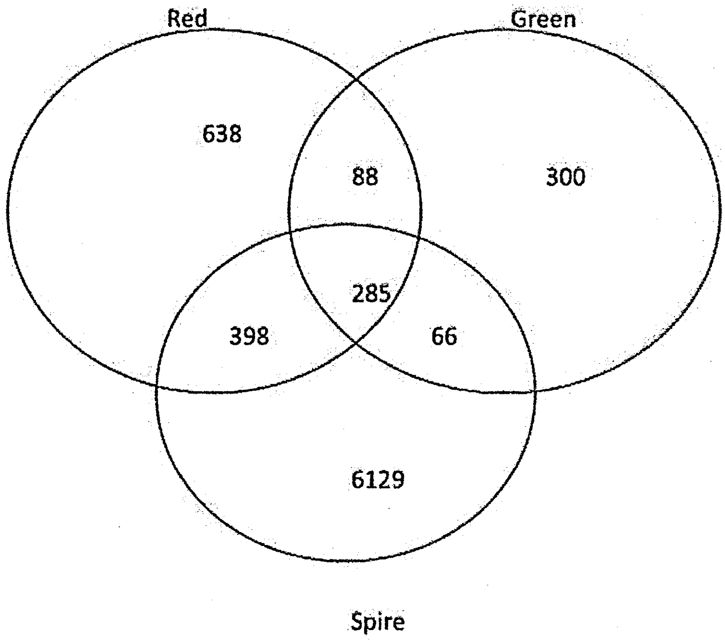


Figure 4.1: This is a Venn diagram of the number of sources in the PACS 6σ catalogues in the early science data phase of the Herschel mission generated by the first version of the PACS pipeline programme.

each source into pixel locations. The size of the postage stamp is set such that it is a square with its centre at the source location and its sides are fifty pixels in length. The astrolib routine hextract is used to generate a new header so the postage stamp can be generated. Then each postage stamp is saved as a separate image in a suitably named graphics file.

Then each postage stamp was closely examined, for each source at all three SPIRE wavelengths. The SPIRE 250μm, 350μm and 500μm wavelength images for each source seen in the PACS instrument but not seen in the SPIRE instrument were directly compared. Those postage stamps where the image was a blank field or partially over the edge of the SPIRE field were eliminated. This effect is due to the fact that the PACS and SPIRE instruments are pointing in slightly different directions. These effects reduced the original sample of eighty eight sources down to sixty one sources.

The next stage of source pruning was to exclude those sources where the whole postage stamp was entirely in the SPIRE field but too close to the edge of the field to be reliable. This was done because sources near the edge of the field have unrealistically high fluxes because of the under-sampling or only partial sampling of the PSF. This process reduced the sample of sources that are likely to be reliable down to six.

An IDL programme, that I wrote myself, was used to convert to decimal degrees for all six sources, and the results are saved to an .input file. This programme is necessary because many of the subsequent programmes need the .input file as an input, making use of the in house idl routine readcols.

After that I wrote an IDL programme to generate the Spectral Energy Distribution, SED, and postage stamps for the PACS green and red wavelengths, plus postage stamps for all three SPIRE wavelengths, for all six sources, that are seen in PACS but not seen in SPIRE, in the fully pruned list. If any of the fluxes is negative or zero it is set to $1 \times 10^{-4} Jy$. This is necessary because of the logarithmic vertical scale. For each point in the SED the error bars are plotted on top of each point centred on that point that apply to that point. If the error bar is greater than the flux of a given point in the SED, then a downward facing arrow is printed at that point.

Next an IDL programme was written by myself to help determine whether these six sources seen in PACS but not seen in SPIRE, fully pruned list, are genuine sources. This programme generated for each of the six sources seen in PACS but not seen in SPIRE, fully pruned list, a SED and individual postage stamps for the separate 70 and 71 observations. The SED of M82, is over plotted on to the SED.

The signal to noise of a source, not flux, is used to construct a postage stamp. This is done because the signal to noise is used here to detect point sources. The 70 and 71 in the PACS data set names specifies the scan direction used in constructing the image. The 70 and 71 scan directions are perpendicular to each other. This helps to detect bogus sources because if a source appears at one location in one scan direction but not in the other, then that source can not be genuine. Even if a source appears at the same location in both scan directions does not guarantee that the source is genuine, but is a good indication.

The six sources found in both the PACS green and red wavebands but not seen in the SPIRE wavebands, fully pruned list, are labeled one to six.

None of the six sources have identifications in SIMBAD, Set of Identifications Measurements and Bibliography for Astronomical Data. SIMBAD is a database of objects principally lying inside the solar system or stars inside are own Galaxy. Two of the six sources do not have identifications in NED, NASA Extra Galactic Data base. The other four sources have identifications that correspond to known galaxies. The NED database includes a large number of galaxies plus other extra galactic objects.

The NED search radius is 0.192 arc minutes. The search radius is the resolution of the PACS red wave band in arc minutes. The search radius I used in the SIMBAD coordinate query was 0.19 arc minutes.

I will give a brief summary of the six sources in section 5.1., one paragraph for each source, starting with the two catalogue numbers first, and then the right ascension and declination and finally any other relevant information. Table 4.1 gives information on the six sources.

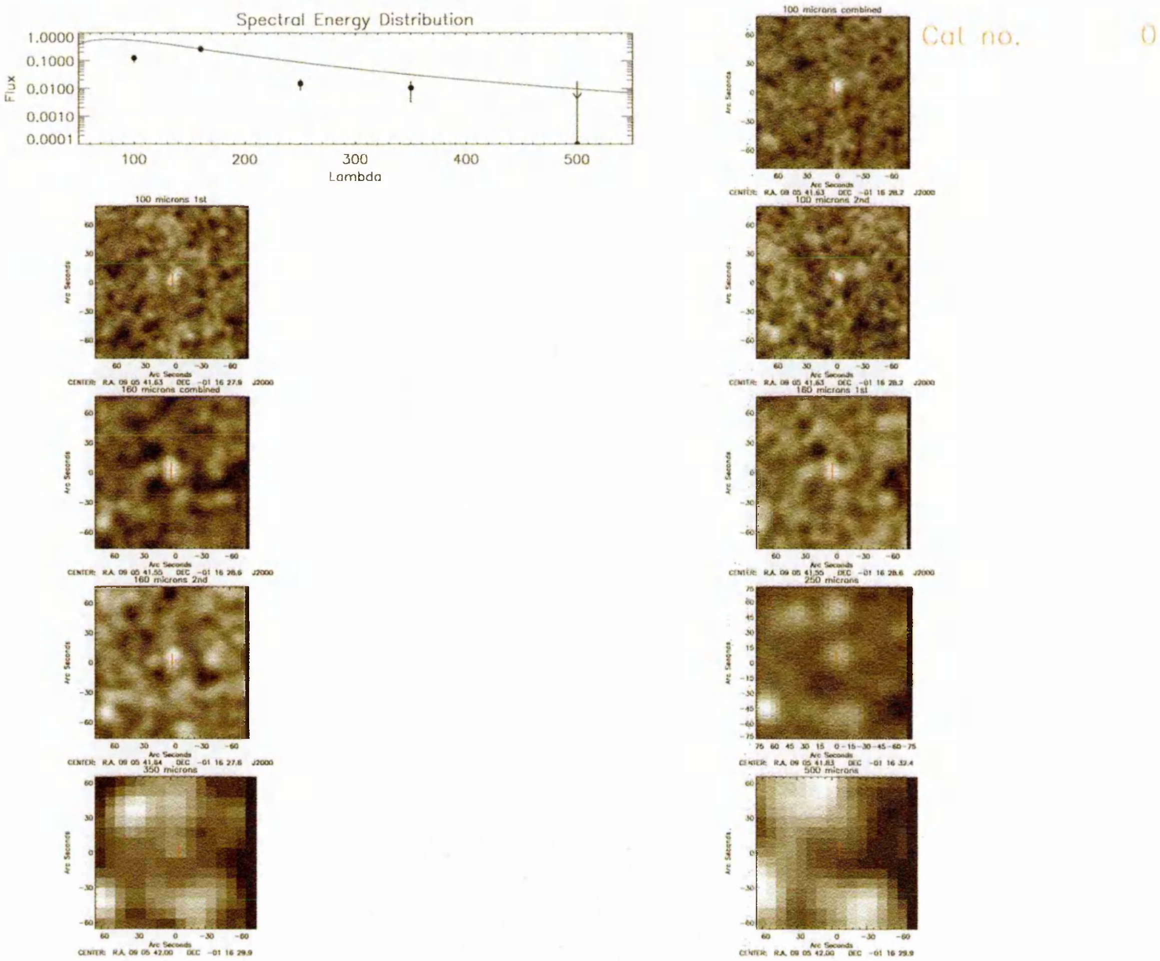


Figure 4.2: This figure shows the SED and the postage stamps for source catalogue number 0 in the six sources seen in PACS but not in Spire catalogue. It also shows postage stamps of both scan directions for the PACS instrument. The SED of M82 the famous starburst galaxy is also shown.

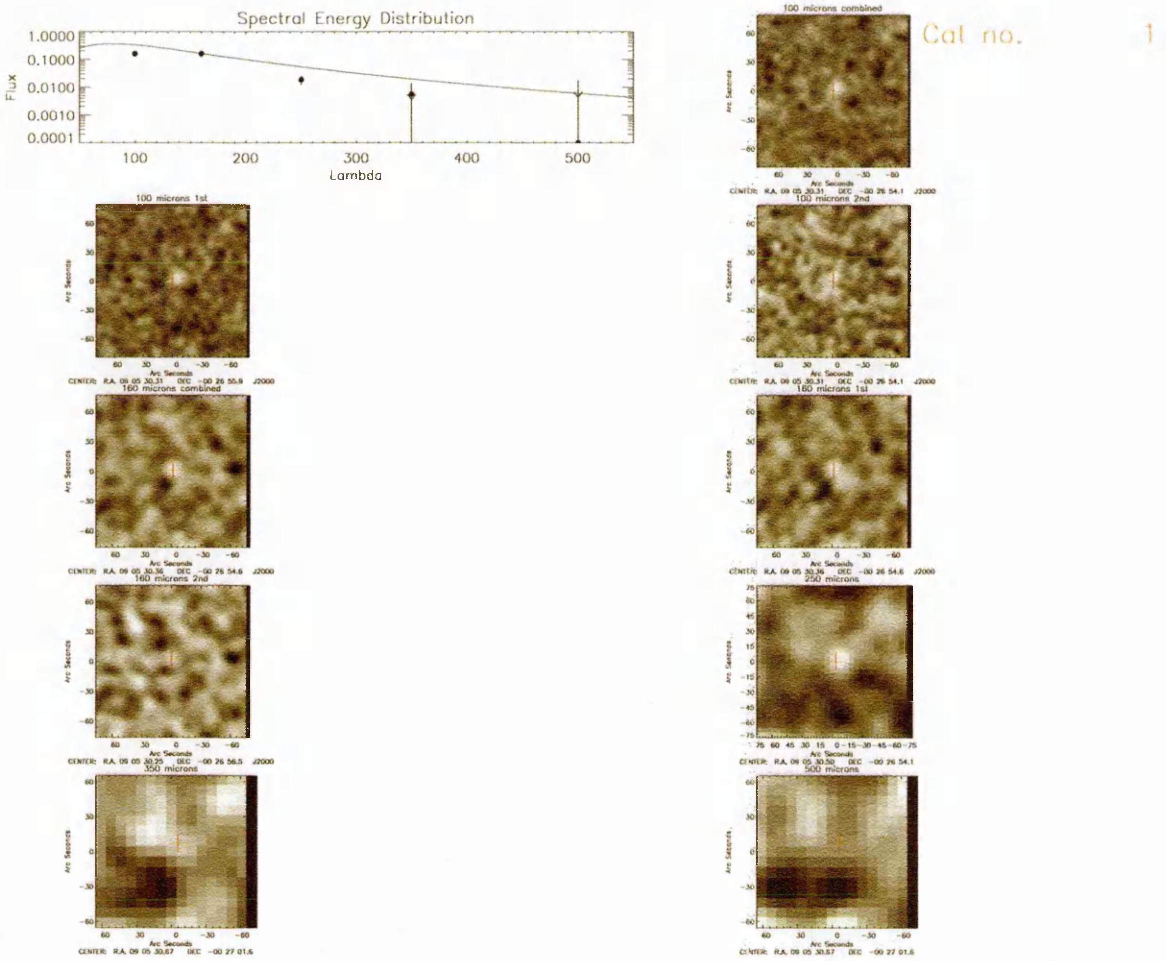


Figure 4.3: This figure shows the SED and the postage stamps for source catalogue number 1 in the six sources seen in PACS but not in Spire catalogue. It also shows postage stamps of both scan directions for the PACS instrument. The SED of M82 the famous starburst galaxy is also shown.

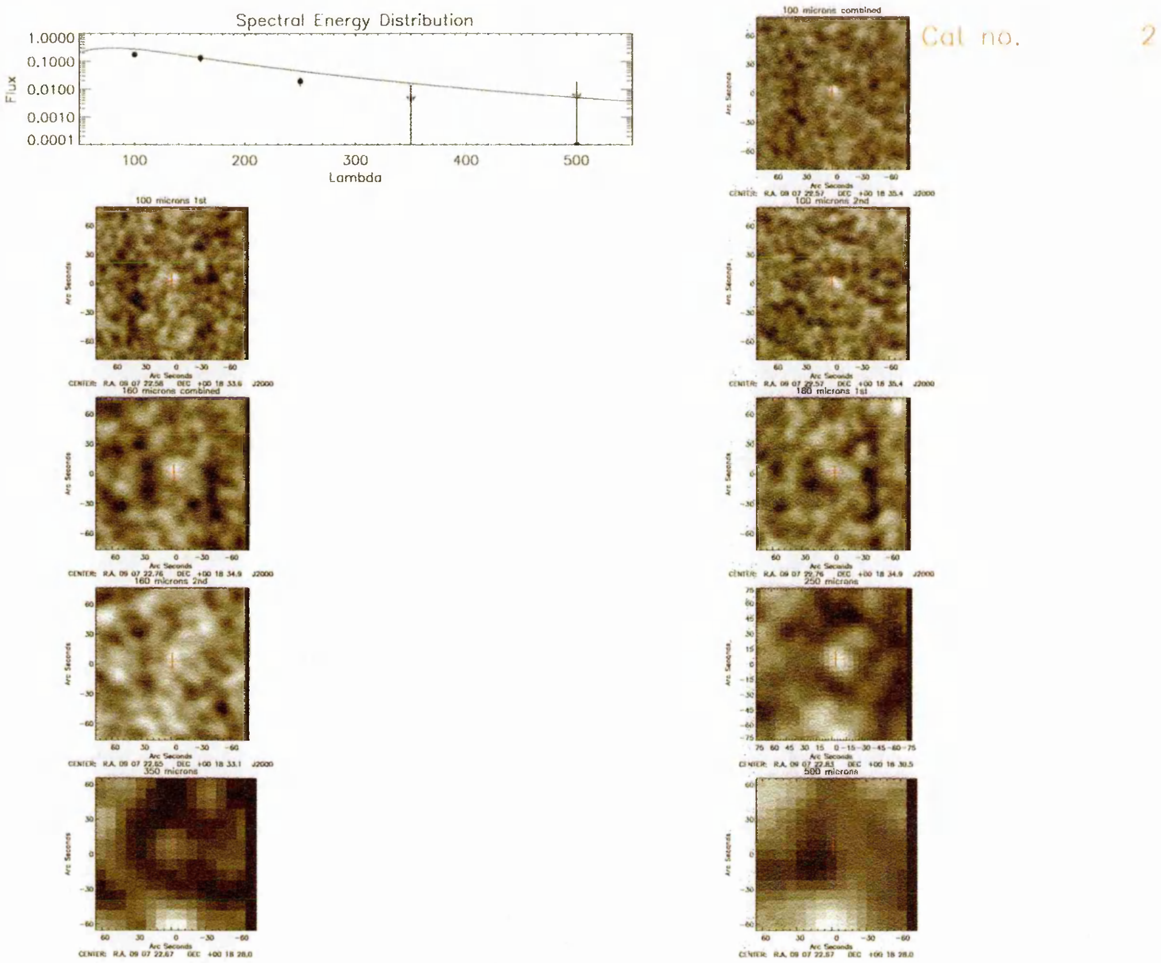


Figure 4.4: This figure shows the SED and the postage stamps for source catalogue number 2 in the six sources seen in PACS but not in Spire catalogue. It also shows postage stamps of both scan directions for the PACS instrument. The SED of M82 the famous starburst galaxy is also shown.

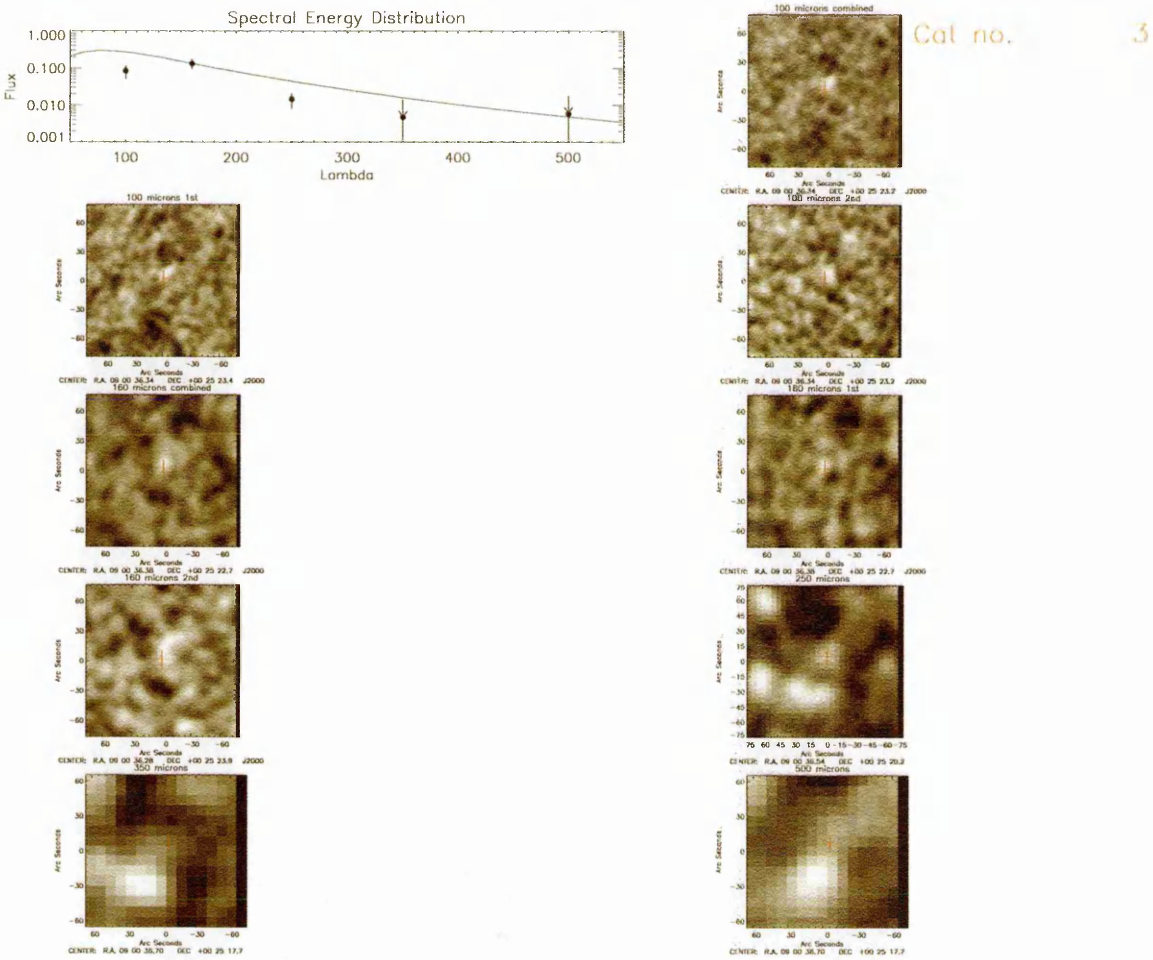


Figure 4.5: This figure shows the SED and the postage stamps for source catalogue number 3 in the six sources seen in PACS but not in Spire catalogue. It also shows postage stamps of both scan directions for the PACS instrument. The SED of M82 the famous starburst galaxy is also shown.

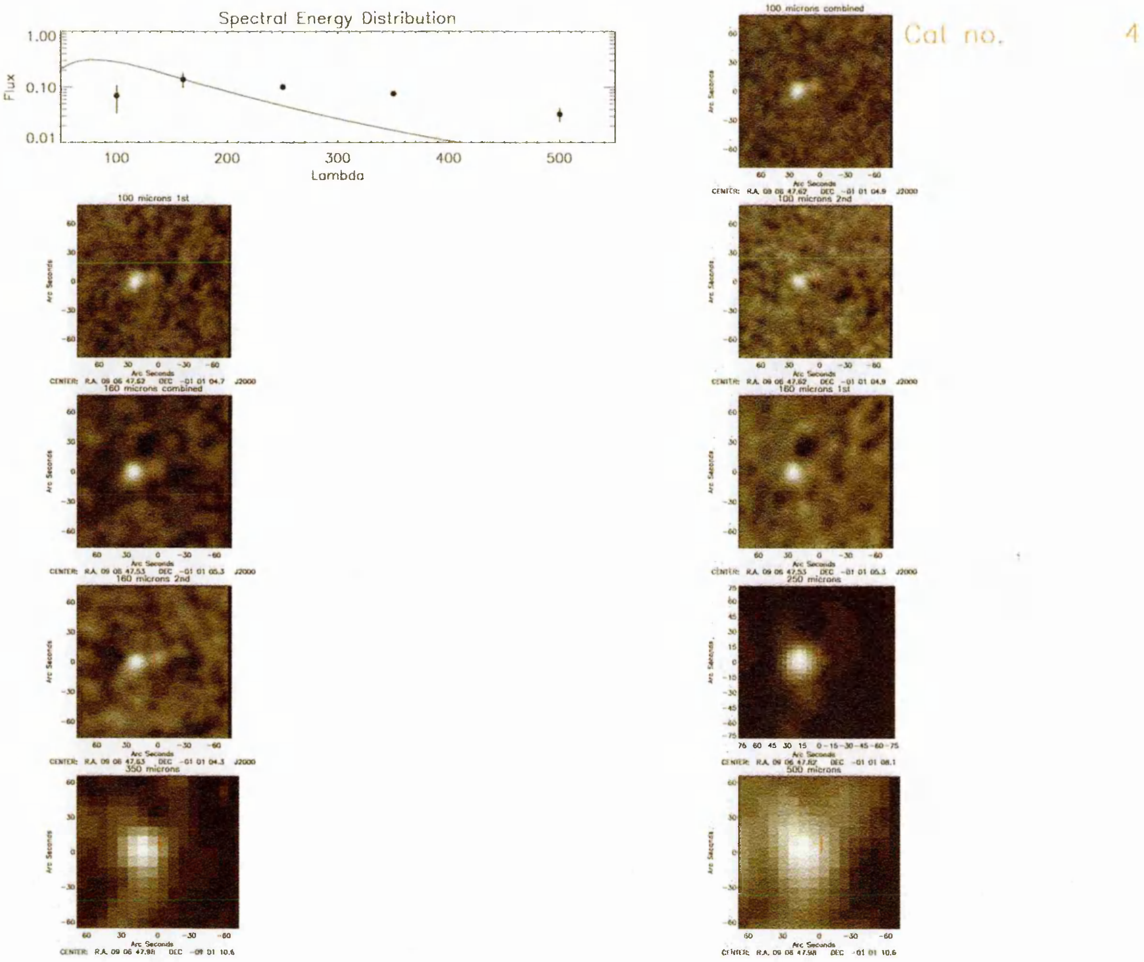


Figure 4.6: This figure shows the SED and the postage stamps for source catalogue number 4 in the six sources seen in PACS but not in Spire catalogue. It also shows postage stamps of both scan directions for the PACS instrument. The SED of M82 the famous starburst galaxy is also shown.

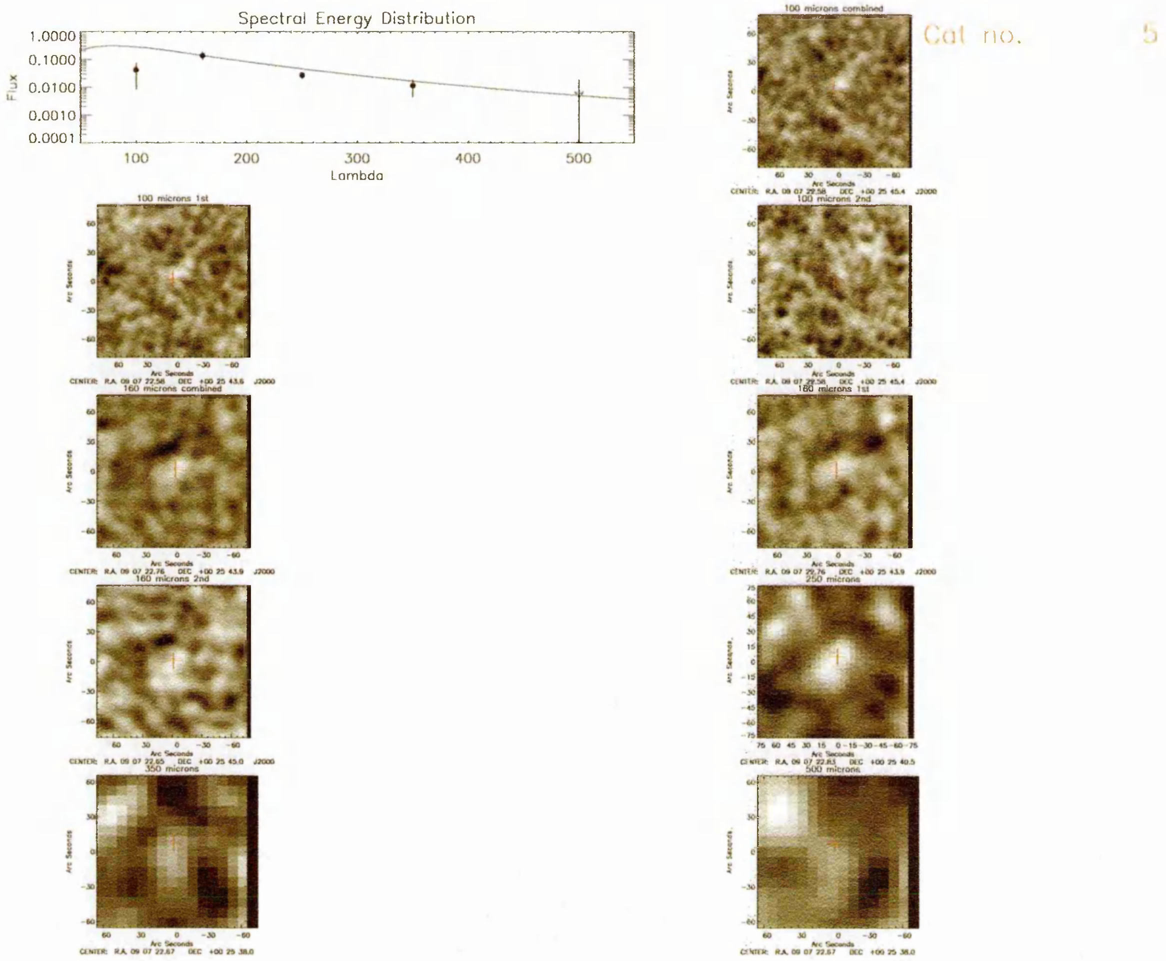


Figure 4.7: This figure shows the SED and the postage stamps for source catalogue number 5 in the six sources seen in PACS but not in Spire catalogue. It also shows postage stamps of both scan directions for the PACS instrument. The SED of M82 the famous starburst galaxy is also shown.

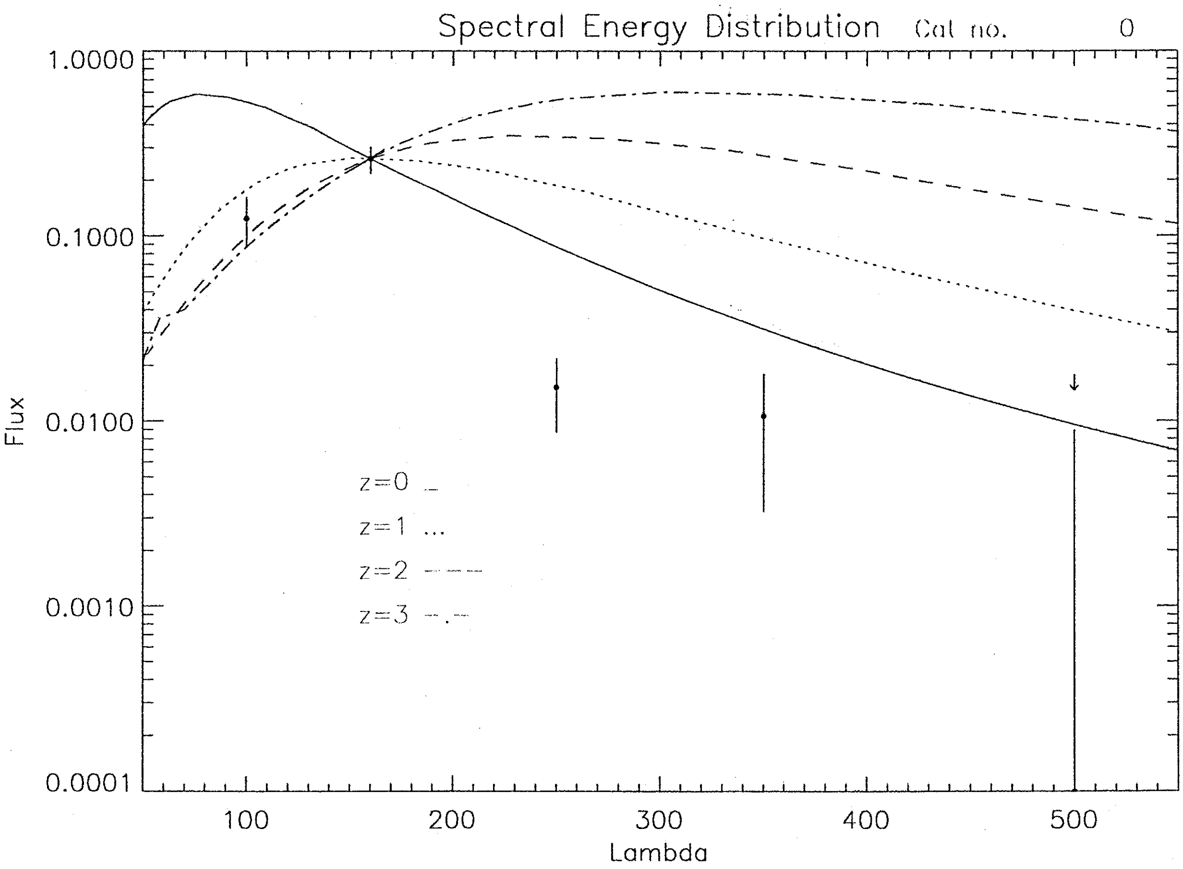


Figure 4.8: This figure shows the SED for source catalogue number 0. The SED of M82 is also shown for various redshifts. The Lambda scale is in microns.The flux is in Janskys. The 500 micron flux is negative so it has been set to 0.1mJy. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

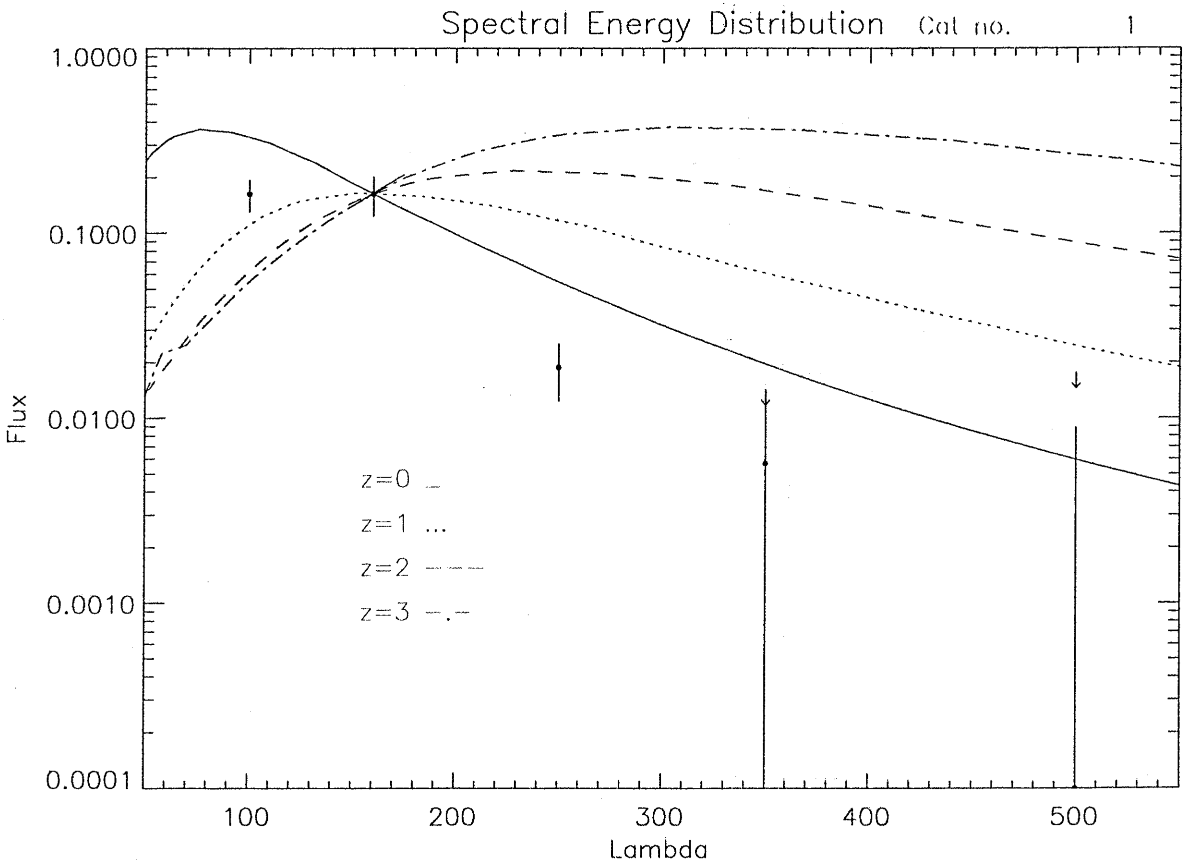


Figure 4.9: This figure shows the SED for source catalogue number 1. The SED of M82 is also shown for various redshifts. The Lambda scale is in microns. The flux is in Janskys. The 350 and 500 micron fluxes are both negative so they have both been set to 0.1mJy. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

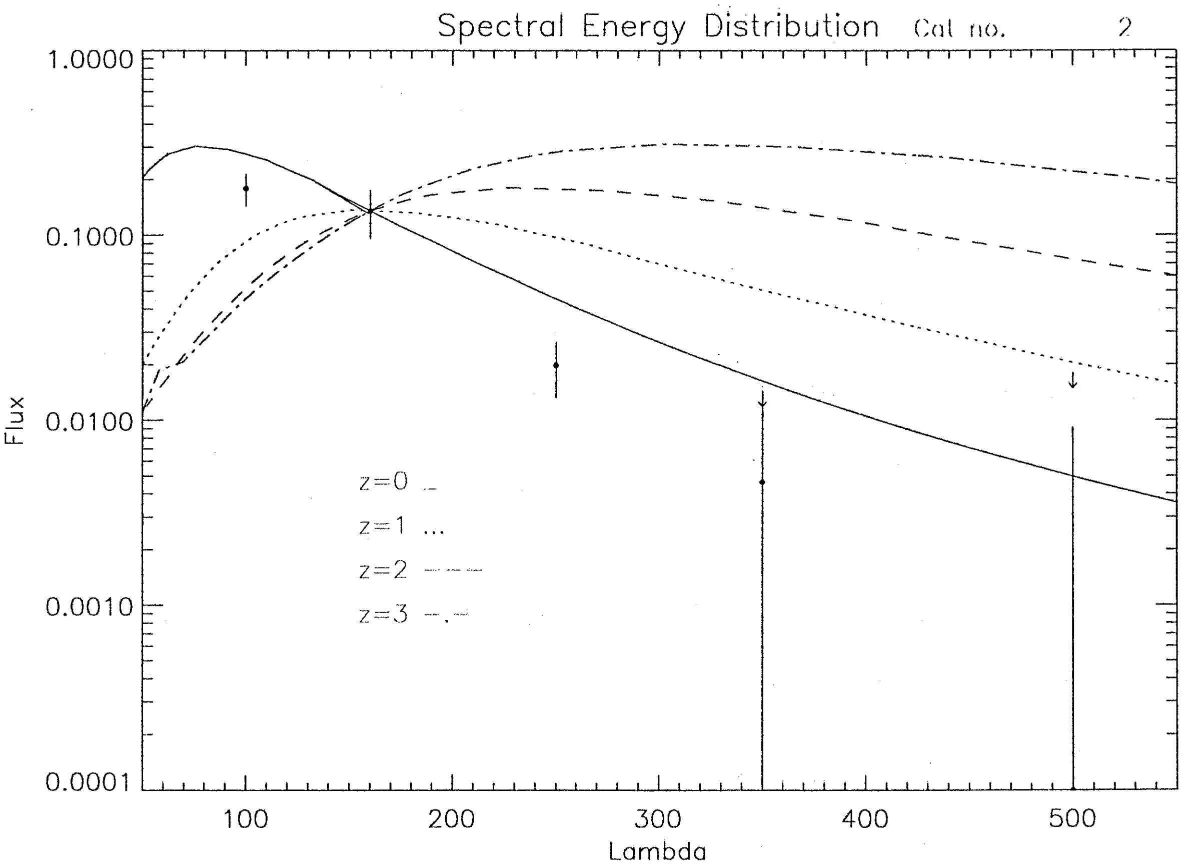


Figure 4.10: This figure shows the SED for source catalogue number 2. The SED of M82 is also shown for various redshifts. The Lambda scale is in microns. The flux is in Janskys.The 350 and 500 micron fluxes are both negative so they have both been set to 0.1mJy. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

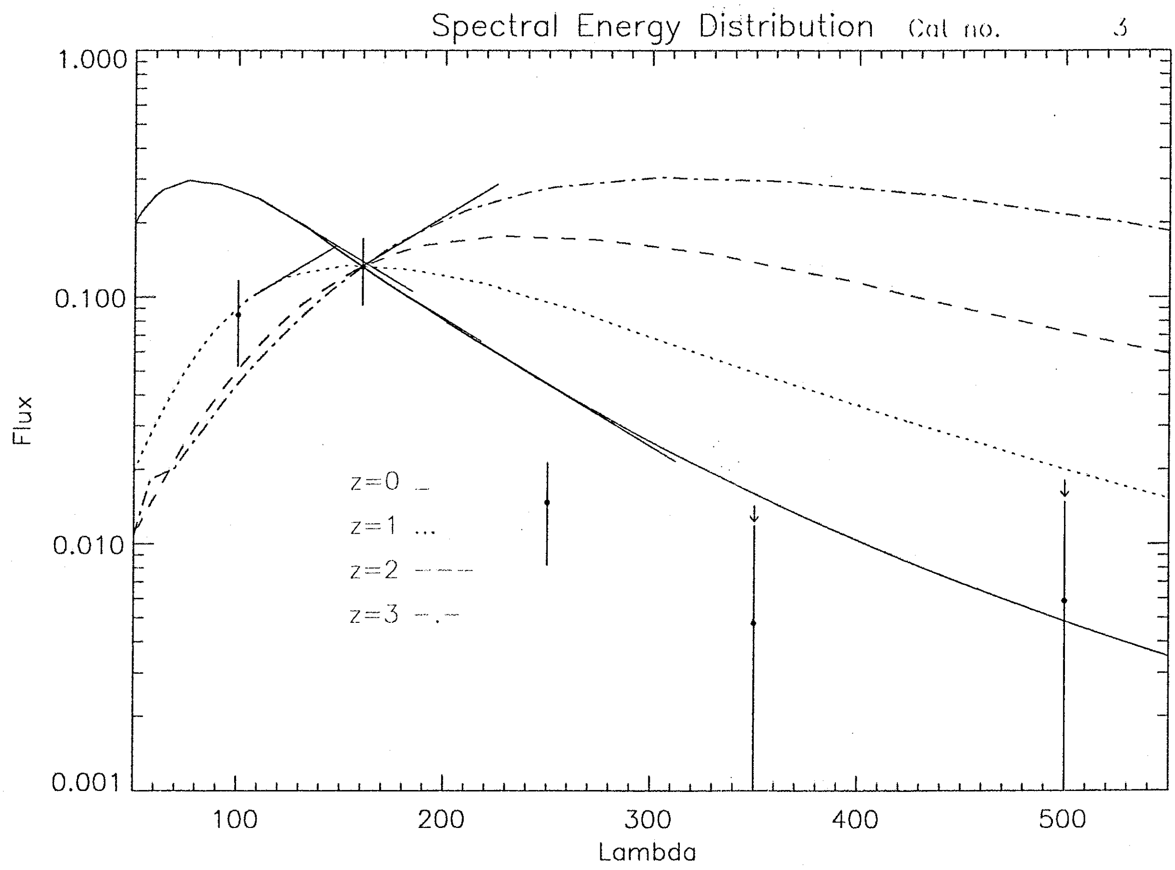


Figure 4.11: This figure shows the SED for source catalogue number 3. The SED of M82 is also shown for various redshifts. The Lambda scale is in microns. The flux is in Janskys. The 350 and 500 micron fluxes are both negative so they have both been set to 0.1mJy. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

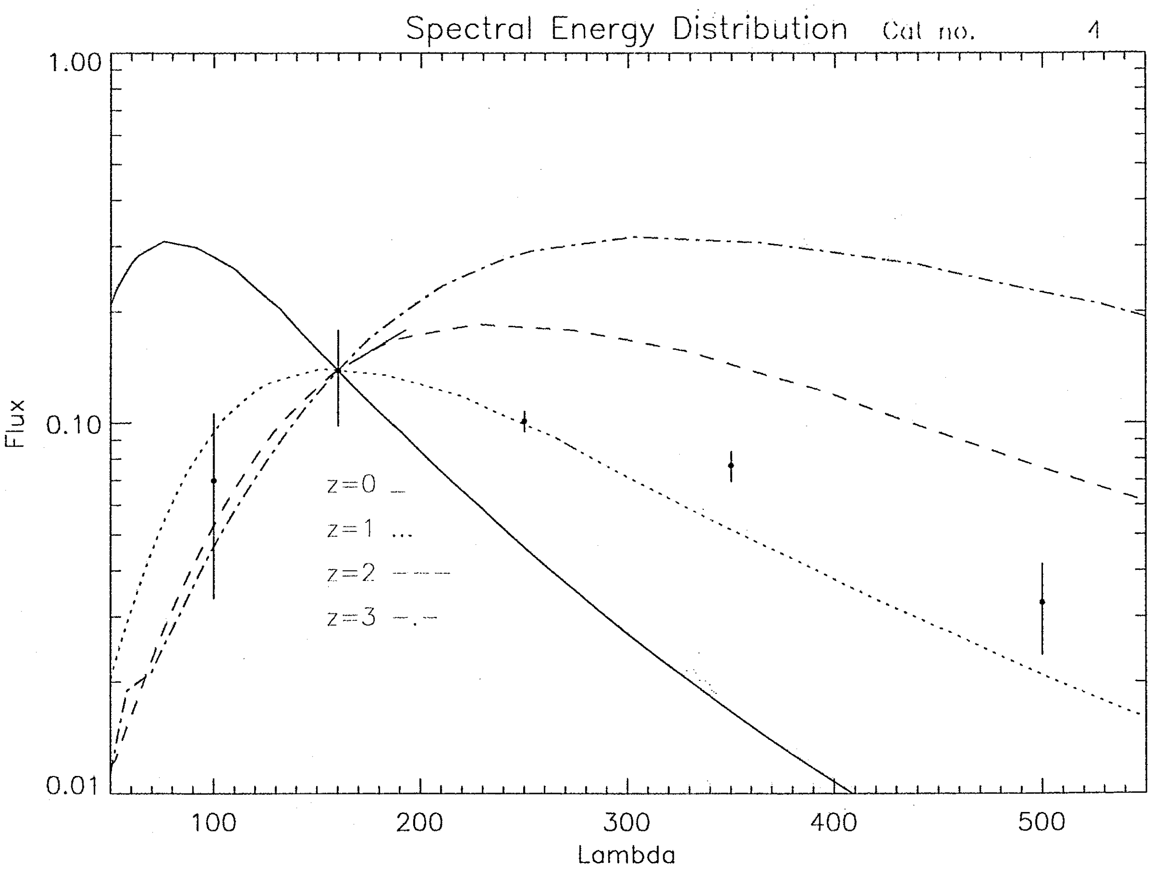


Figure 4.12: This figure shows the SED for source catalogue number 4. The SED of M82 is also shown for various redshifts. The Lambda scale is in microns. The flux is in Janskys. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

Catalogue numbers	Ra	Dec	Number of NED IDs
1,27	09:05:41.57	-01 16 27.4	0
2,28	09:05:30.24	-00 26 54.1	3
3,36	09:07:22.57	00 18 35.5	1
4,48	09:00:36.27	00 25 25.0	1
5,53	09:06:47.55	-01 01 3.1	0
6,57	09:07:22.57	00 25 45.5	1

Table 4.1: The summary of the data of the six sources.

4.4.2 The PACS-only sources for the third version of the PACS pipeline

A second set of PACS only sources was obtained from the output data of the third version of the PACS pipeline programme, using the same early science phase Herschel data as the first set of PACS only sources. The same basic procedure was used in generating the second set of PACS only sources as the first. The only difference was that different data files were used plus different IDL programmes were used to process the data. These new programmes were basically the same as the programmes used to generate the first set of PACS only data.

The source catalogues of the PACS red and green sources used in generating the second set of PACS only sources used the connected_pixels, the in house IDL routine, with a 5σ signal to noise threshold. There are only fifty three sources that are seen in both the second PACS red and green source catalogues but not seen in the SPIRE instrument. Postage stamps for all fifty three sources seen in the second PACS red

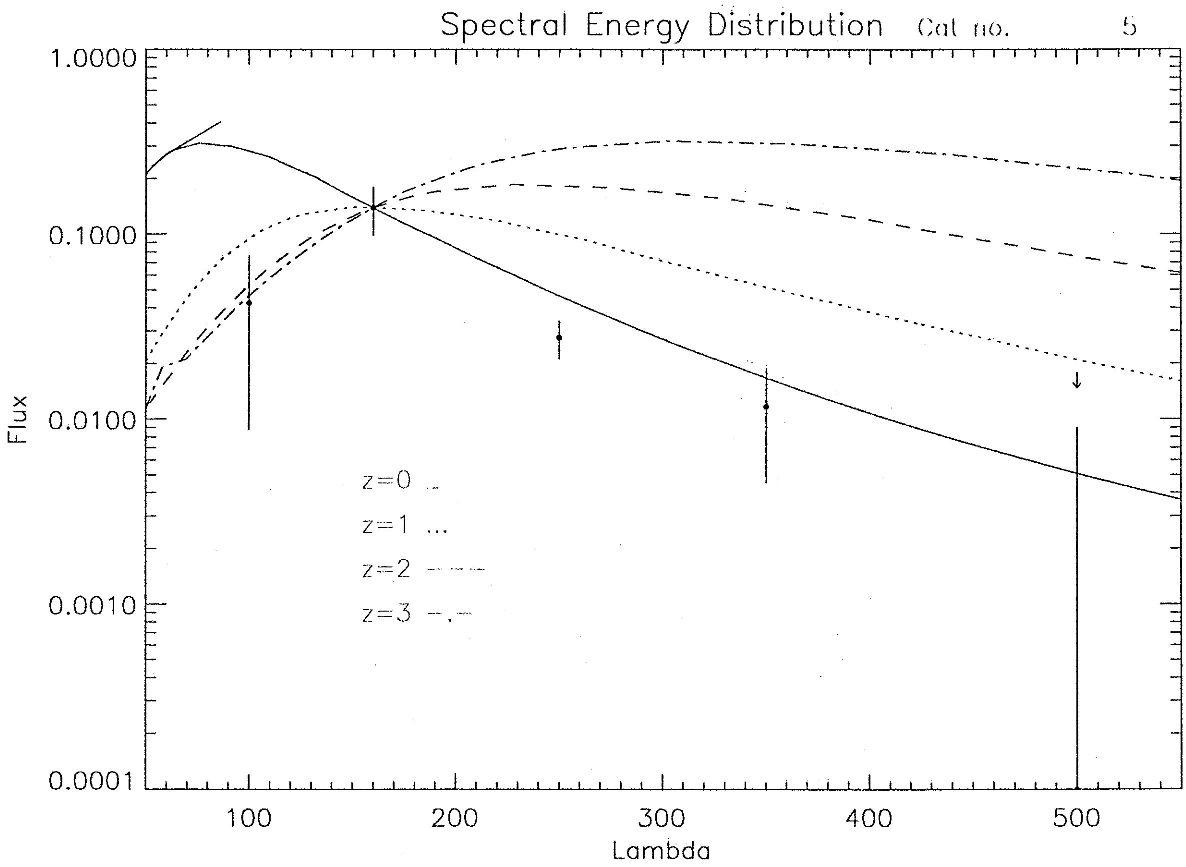


Figure 4.13: This figure shows the SED for source catalogue number 5. The SED of M82 is also shown for various redshifts. The Lambda scale in microns. The flux is in Janskys. The 500 micron flux is negative so it has been to 0.1mJy. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

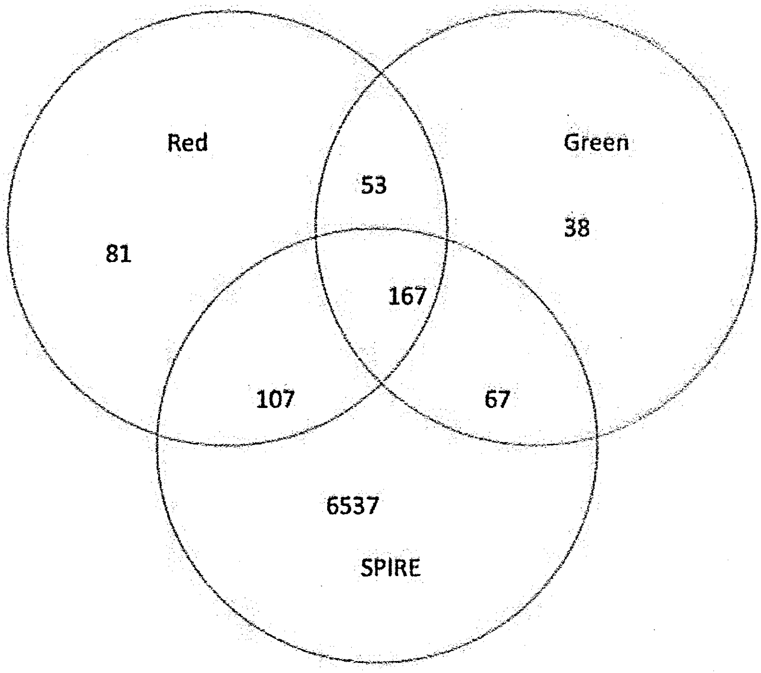


Figure 4.14: This is a Venn diagram of the number of sources in the PACS 5σ catalogues in the early science data phase of the Herschel mission generated by the third version of the PACS pipeline programme.

and green source catalogues but not seen in the SPIRE instrument, at the SPIRE 250μm, 350μm and 500μm wavelengths were generated. There are three such postage stamps for each source seen in the second PACS red and green source catalogues but not seen in the SPIRE instrument.

Then each postage stamp image was closely examined, for each source at all three SPIRE wavelengths. The SPIRE 250μm, 350μm and 500μm wavelength images for each source seen in the second PACS red and green source catalogues but not seen in the SPIRE instrument, were directly compared. Those postage stamps where the image was a blank field or partially over the edge of the SPIRE field were eliminated. These effects reduced the original sample of fifty three sources down to one source. This process is referred to as pruning.

The catalogue numbers of the sources seen in the second PACS red and green source catalogues but not seen in the SPIRE instrument run from zero to fifty two. Although this source is not seen in SPIRE, I am using the two positions given in the postage stamps of the centre of the field. There are two sets of right ascension and declination for this source, one from both the SPIRE 350 and SPIRE 500 postage stamps, the other from the SPIRE 250 postage stamp. These positions are very close on the sky, and referred to as A and B respectively. The A and B values of the right ascension and declination of the source that survived the pruning process is due to rounding errors plus the different pixel sizes of the SPIRE 250, 350 and 500 images. This is due to their different headers. Using the A and B sets of right ascension and declination, and the astrolib routine gcirc, the great circle distance on the sky between the two source positions was obtained. This is equal to 3.6".

4.4.3 The only source found in PACS not in SPIRE using the final version of the PACS pipeline

Postage stamps were generated of all six sources found in the first PACS red and green source catalogues but not seen in the SPIRE instrument that survived the selection process, using the output data of third version of the PACS pipeline programme. See figure 4.15..There clearly is a source at the centre of each postage stamp, marked by a red cross. It is just too weak to pass the 5σ threshold. Additionally the SNR may have been degraded by the SNR penalty for white noise in the timeline filtering part of the third version of the PACS pipeline programme.

No objects were found in SIMBAD with both A and B sets of right ascension and declination using a search radius of 0.19 arcminutes. No objects found in NED with both A and B sets of right ascension and declination using a search radius of 0.192 arcminutes. These search radii were chosen because they are the closest option in NED and SIMBAD to the resolution of the Herschel space observatory for the PACS red waveband.

The resolution of the PACS red waveband is 11.5", so a great circle separation of 3.6" between the A and B position of the source, is well within the positional uncertainties from pixellisation.

The source location A has a right ascension of 09:05:41.33 and a declination of -01°16'29.9". The source location B has a right ascension of 09:05:41.50 and a declination of -01°16'27.4". Since the A and B positions of the source is a plotting centering artifact, I only given figures for position A.

4.5 QSO stacks

I was given by my supervisor a, .txt file, containing a list of 411 quasar candidates along with additional information about them. The quasars were selected from the SDSS as stellar objects, with blue colours, then followed up with SDSS spectra. In addition, it contains quasars from the 2SLAQ survey. More details on the quasar compilation can be found in Serjeant et al. 2010. (A&A 518 L7). For each candidate there are eight pieces of information such as: its right ascension, its declination, its absolute magnitude, its redshift, its SPIRE flux, its SPIRE flux error plus the SPIRE wavelength it has been seen in. The SPIRE wavelengths can only take three values, 250μm, 350μm and 500μm.

Then I wrote an IDL programme that generates the 100μm and 160μm, stacked

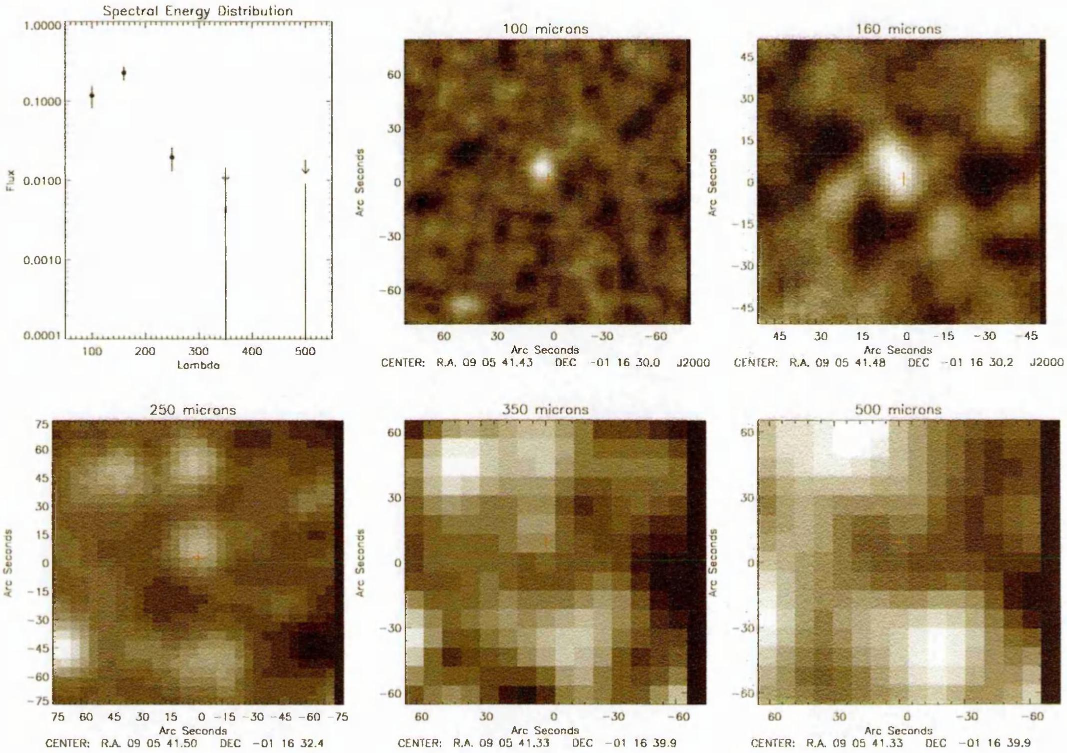


Figure 4.15: This figure shows the SED and postage stamps for source location A.

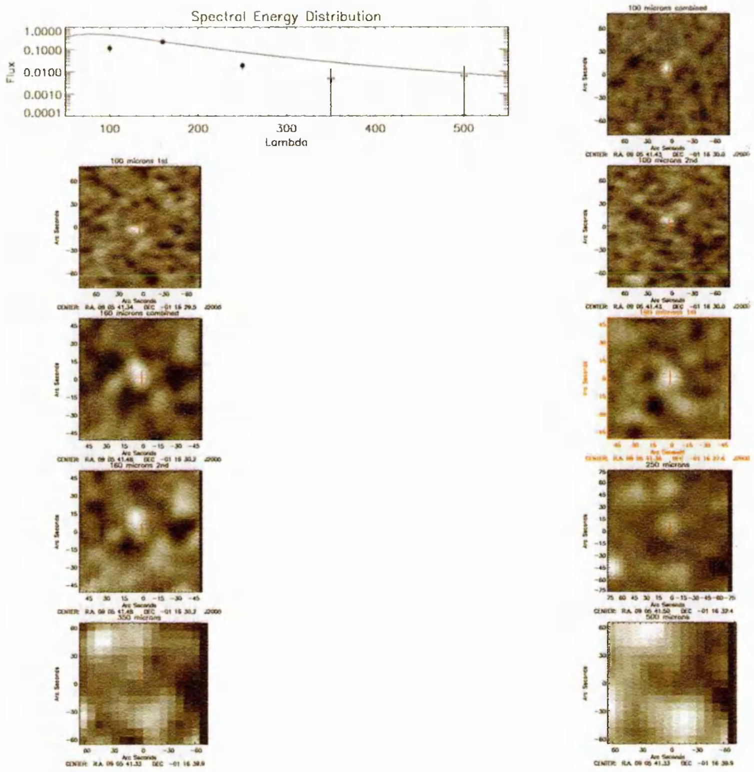


Figure 4.16: This figure shows the SED and postage stamps for source location A. The SED of M82 is also shown. Postage stamps are also shown for both PACS red and green wavebands and for both scan directions.

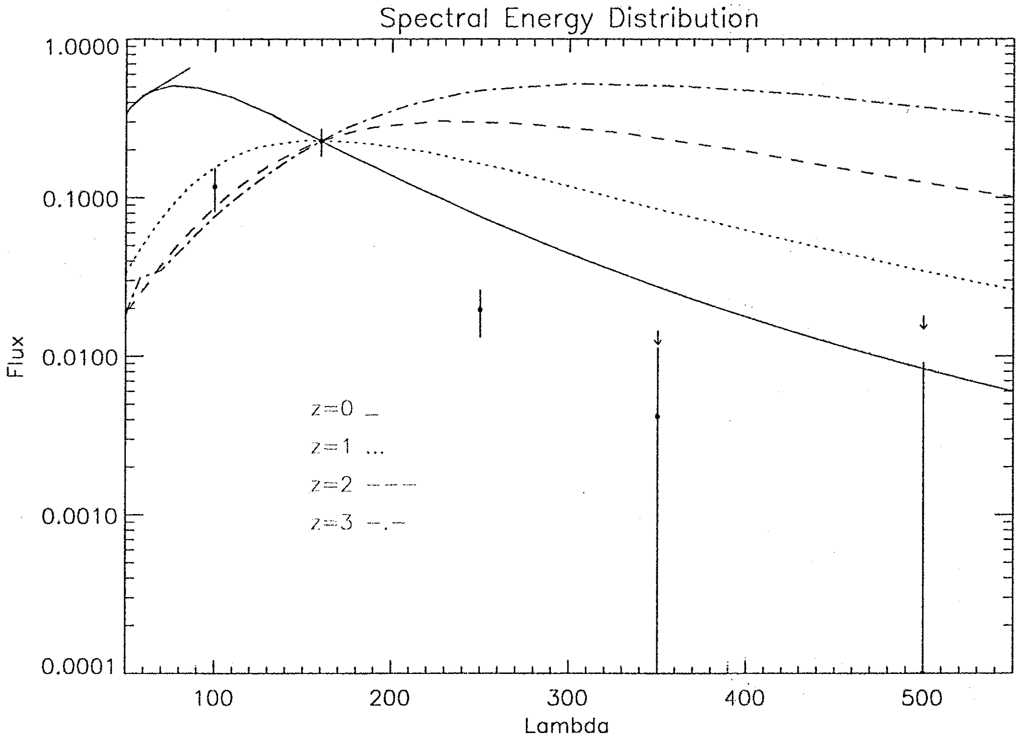


Figure 4.17: This figure shows the SED for source location A. The SEDs of M82 are also shown for different redshifts. The Lambda scale in microns. The flux scale is in Janskys. The 350 and 500 micron fluxes are both negative so they both have seen set to 0.1mJy. All fluxes are plotted with 1-sigma error bars except those with fluxes that have been set to 0.1mJy, which have an upper limit of 2-sigma.

fluxes of all the 411 quasar candidates plus the uncertainties in the detections. The same programme also generates an IDL save file, containing for each of the 411 quasar candidates, the right ascension, the declination, the 160μm flux, the 100μm flux, the 160μm noise and the 100μm noise for each source.

The noisiest parts of the map are clipped out. The 160μm flux for each source is obtained using the pixel locations of the sources and the mean of the 160μm flux where the 160μm noise is less than its standard deviation is subtracted from the 160μm flux. The 160μm noise is obtained for each source using the pixel locations of the sources.

The same procedure is used for the 100μm data.

Then the mean of the 160μm flux is obtained. After that the mean of the 100μm flux is obtained. The standard error of the 160μm mean flux is evaluated. Next the standard error of the 100μm mean flux is evaluated. The standard error is obtained by dividing the standard deviation by the square root of the number of sources in the catalogue.

The result is: 100μm stacked flux is 2.987±0.093mJy (32σ detection); 160μm stacked flux is 4.7±1.1mJy (4.3σ detection).

After that I wrote two IDL programmes that generate histograms of the number of objects in a given flux range using in-house histogram or overlaying routines.

The inputs to the two programmes are the coadded green70 into green71 flux data, the coadded red70 into red71 flux data and the quasar catalogue data.

The coadded green70 into green71 flux data was read in with its header, the combined signal and the combined noise. Then the array used for generating the histograms of the coadded green70 into green71 flux is obtained by chopping off the edges

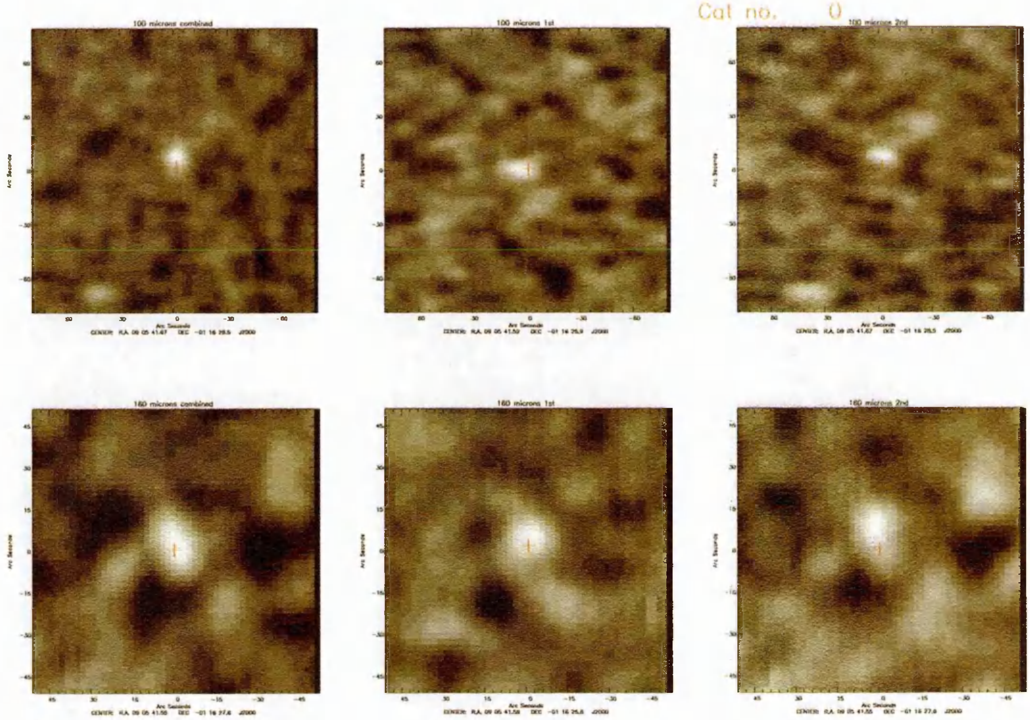


Figure 4.18: This figure shows the postage stamps for source catalogue number 0, of the six sources seen in PACS but not in SPIRE, from the output of the first version of the PACS pipeline programme. The postage stamps are PACS red and green wavebands and for both scan directions. Using the output of the third version of the PACS pipeline programme to generate them.

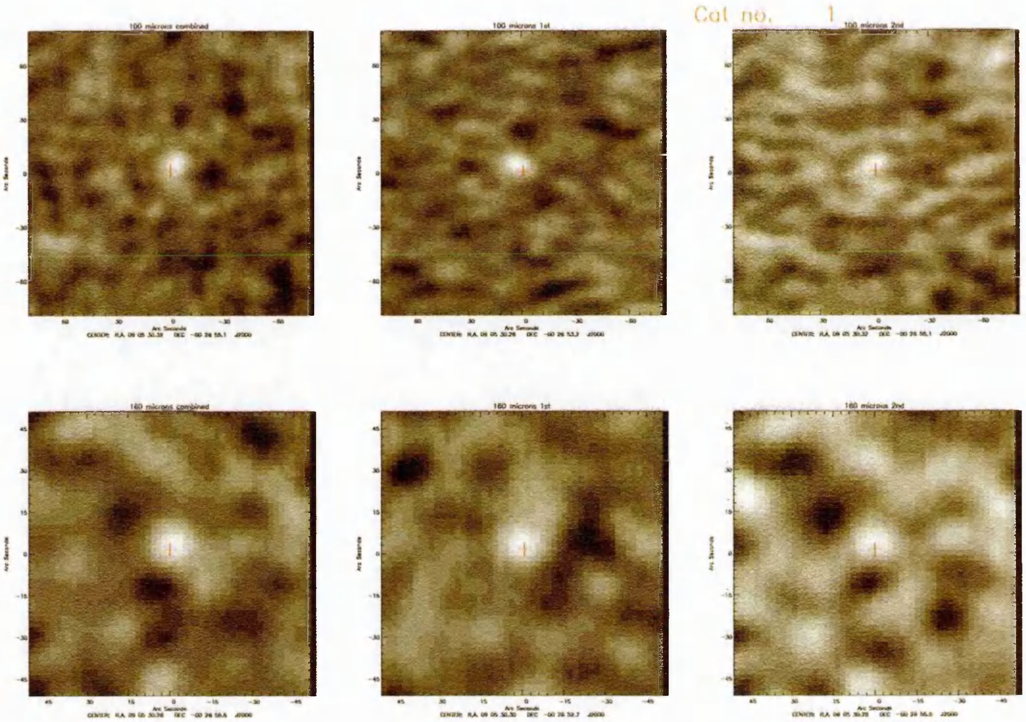


Figure 4.19: This figure shows the postage stamps for source catalogue number 1, of the six sources seen in PACS but not in SPIRE, from the output of the first version of the PACS pipeline programme. The postage stamps are PACS red and green wavebands and for both scan directions. Using the output of the third version of the PACS pipeline programme to generate them.

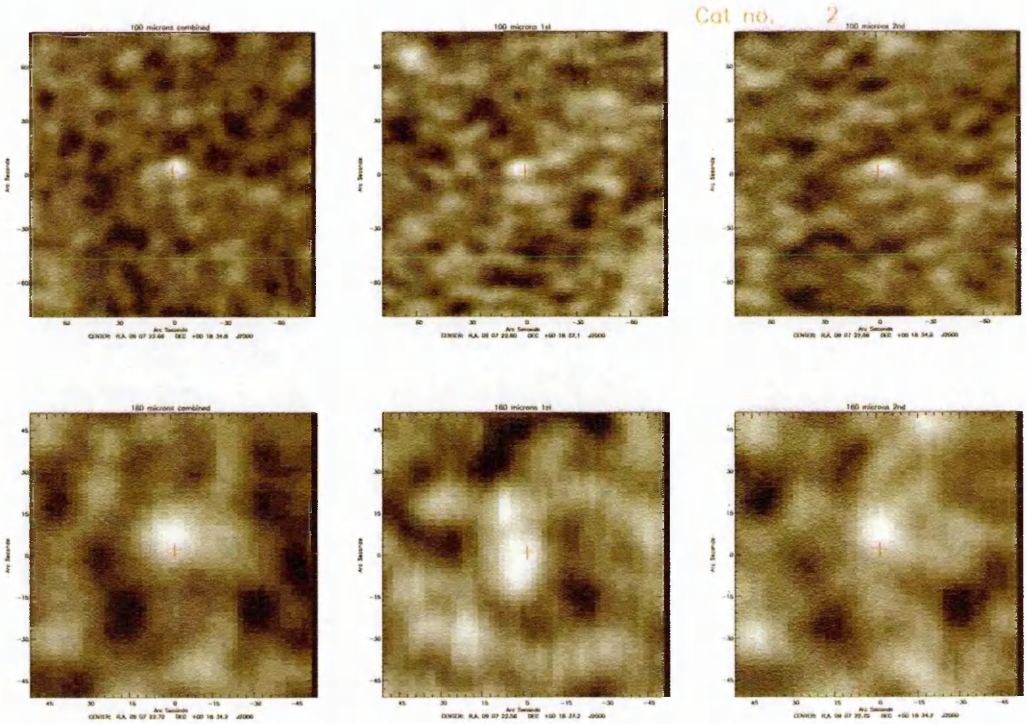


Figure 4.20: This figure shows the postage stamps for source catalogue number 2, of the six sources seen in PACS but not in SPIRE, from the output of the first version of the PACS pipeline programme. The postage stamps are PACS red and green wavebands and for both scan directions. Using the output of the third version of the PACS pipeline programme to generate them.

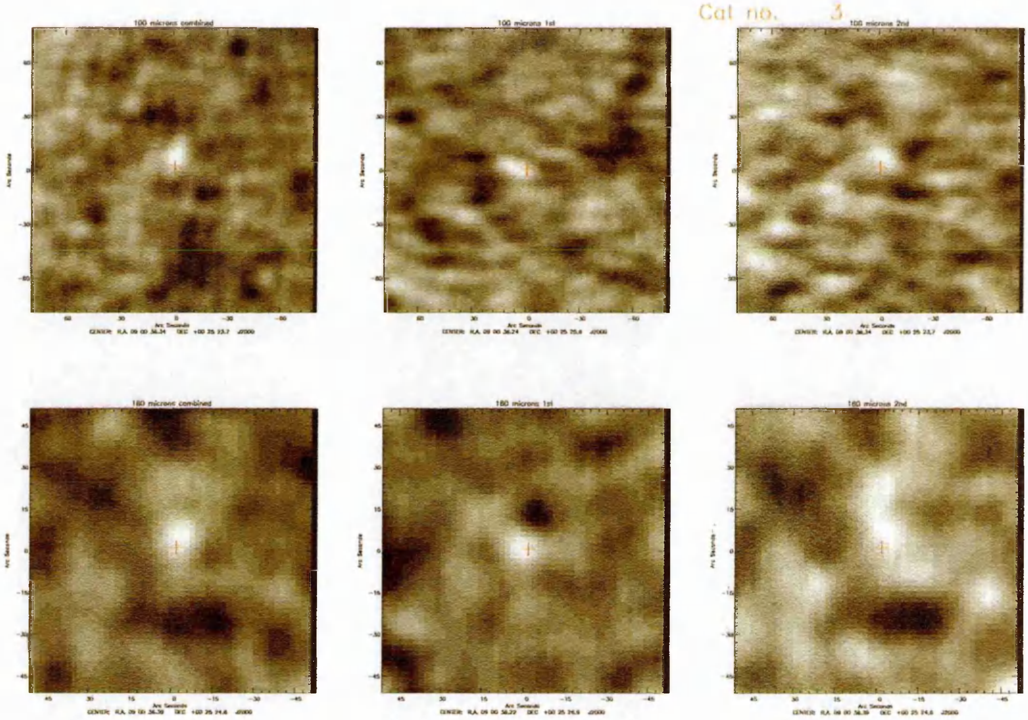


Figure 4.21: This figure shows the postage stamps for source catalogue number 3, of the six sources seen in PACS but not in SPIRE, from the output of the first version of the PACS pipeline programme. The postage stamps are PACS red and green wavebands and for both scan directions. Using the output of the third version of the PACS pipeline programme to generate them.

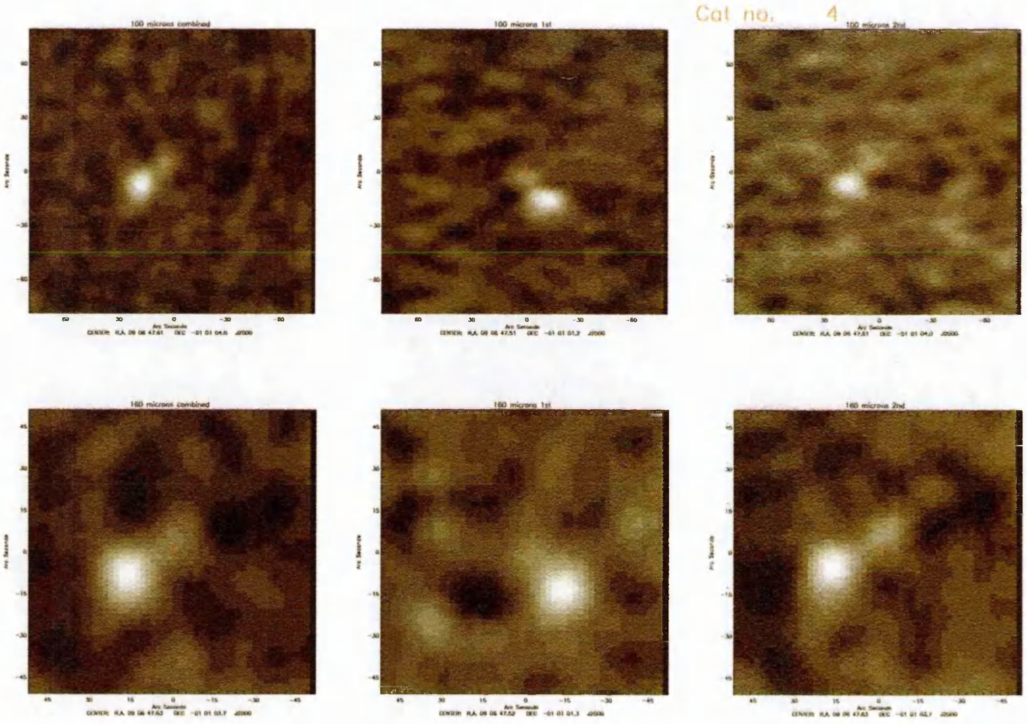


Figure 4.22: This figure shows the postage stamps for source catalogue number 4, of the six sources seen in PACS but not in SPIRE, from the output of the first version of the PACS pipeline programme. The postage stamps are PACS red and green wavebands and for both scan directions. Using the output of the third version of the PACS pipeline programme to generate them.

of the combined signal array. This is done because this data set has unrealistically large fluxes near their edges.

A similar procedure was done with the red data.

The quasar catalogue data was read in and the right ascension and declination of each source was used to get the pixel locations of the corresponding red and green sources, plus both the headers from the coadded green70 into green71 and the coadded red70 into red71 data files. Then using the combined signals, from the coadded green70 into green71 and the coadded red70 into red71 the green and red fluxes for each quasar was obtained. Only those fluxes that were non zero were used in generating the histograms.

The histograms of the green quasar flux plus the coadded green70 into green71 flux were plotted on the same graph and are shown in the figure 4.24.. The hatched bins represent the green quasar fluxes. The unhatched bins represent the coadded green70 into green71 fluxes. The horizontal scale is the green flux. The left hand scale is the number of members of the green quasar flux catalogue in a given bin. The right hand scale is the number of members of the coadded green70 into green71 flux data set in a given bin. The two scales are different.

Figure 4.25. shows the analogous red plots.

The histogram programme also evaluates and prints out the KS, Kolmogorov-Smirnov, probability as a percentage. This is the probability that the two data sets sample the same underlying distribution.

KS_{probg} is the KS probability for the green quasar fluxes compared with the coadded green70 into green71 fluxes. KS_{probr} is the KS probability for the red quasar

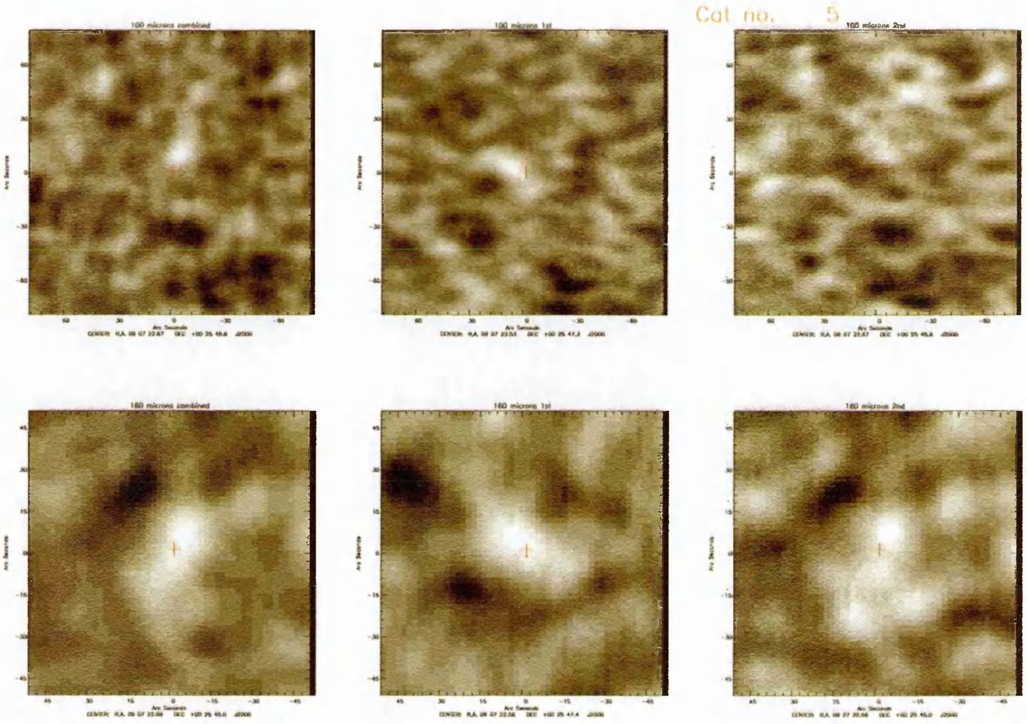


Figure 4.23: This figure shows the postage stamps for source catalogue number 5, of the six sources seen in PACS but not in SPIRE, from the output of the first version of the PACS pipeline programme. The postage stamps are PACS red and green wavebands and for both scan directions. Using the output of the third version of the PACS pipeline programme to generate them.

fluxes compared with the coadded red70 into red71 fluxes:

$$KS_{probg} = 4.66 \times 10^{-6}\% \tag{4.4}$$

$$KS_{probr} = 7.43 \times 10^{-7}\% \tag{4.5}$$

Suppose that the far-infrared fluxes of quasars have the same distribution as the far-infrared map, according to the KS test. That is the two data sets have a high KS probability. What can we conclude about the detection or non-detection of star formation in quasars in this data?

We would have a non-detection because the flux detection would have the same distribution as the far-infrared map, which would not yield evidence of star formation.

Since we have low KS probabilities for both the green and red quasar fluxes compared with the corresponding PACS far infra-red maps, we can say we have detections of star formation in both red and green PACS wave bands for our quasar sample.

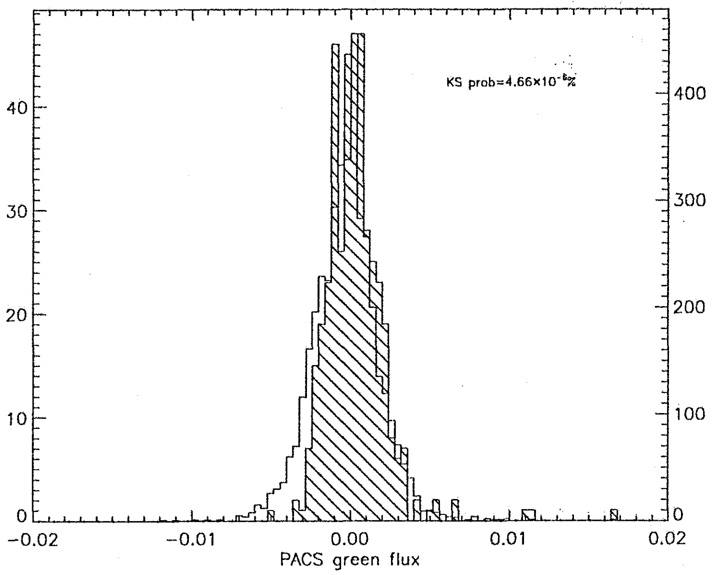


Figure 4.24: This figure shows the histograms of the PACS green quasar flux and the coadded green70 into green71 flux. The horizontal scale is in Janskys.

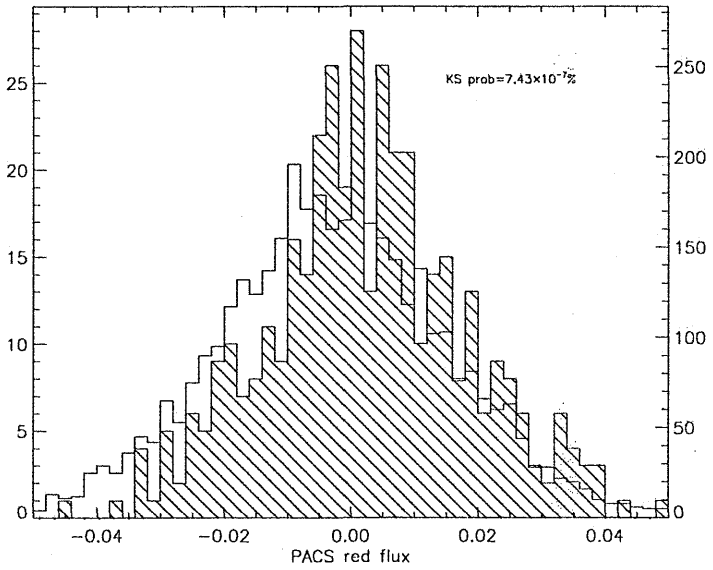


Figure 4.25: This figure shows the histograms of the PACS red quasar flux and the coadded red70 into red71 flux. The horizontal scale is in Janskys.

Chapter 5

Discussion and further work

5.1 Preview of chapter five

In chapter five I will discuss the work I have managed to acheive. Next I will outline the work I would do next if time where no object. Such as looking for evidence for feedback.

5.2 Conclusions of the work so far

The third version of the PACS pipeline programme is working properly. This programme uses the modified kernel, related to the Vio kernel, and the SCUBA matched filter technique to do the timeline filtering. This new approach is doing the timeline filtering, similar to that done in the first version of the PACS pipeline programme, plus spatial filtering in the scan direction. We only need to do the cross-scan filtering with a Gaussian kernel, using the SCUBA matched filter technique, to complete the filtering. In the first version of the PACS pipeline programme after the old timeline filtering has been done to remove $\frac{1}{f}$ noise, two dimensional spatial filtering was done, using a Gaussian kernel, with the SCUBA matched filter.

The FT of the Vio kernel is equal to the PSF, usually represented by the FT of a Gaussian with the instrumental full width at half maximum, divided by the power frequency spectrum of the noise. This may be deduced by using the calculus of variations with Lagrangian undetermined multipliers to maximize the signal to noise subject to the constraint that a source at the origin of known amplitude the signal is equal to that amplitude. See section 3.1.1. for details. The FT of the modified kernel is equal to the PSF divided by the square root of the power frequency spectrum of the noise. This was done to whiten the power frequency spectrum of the filtered timeline to improve point source detection reliability as opposed to optimizing the signal to noise as in the Vio case.

In section 3.1.4. I derived an analytic expression for the ratio of the signal to noise obtained using the modified kernel to the signal to noise obtained using the Vio kernel. This was the basis of an IDL programme I wrote to obtain this ratio for each pixel of the four data sets I have access to. These are red70, red71, green70 and green71. The results of this programme for all four data sets, plus a discussion of the programme is given in section 3.1.4. The overall minimum value of the ratio for all four data sets is 0.608. The overall maximum of the ratio for all four data sets is 0.703. Therefore the signal to noise penalty for white noise is from about 30% to 40%.

As is fully outlined in section 4.3., we have found six sources seen in PACS that are not seen in SPIRE, in the first reduction of the PACS pipeline programme, seen at 6σ. All six sources are seen in both scan directions for both PACS red and green images, so they are likely to be genuine. All six sources are there in the second reduction of the PACS pipeline programme, but are not above 5σ. One further source is seen at greater than 5σ in the PACS image but not seen in the SPIRE image. All the above is outlined in section 4.3. I will now use the numbers I gave these sources in section 4.3.

Source 1, has no identification in NED or SIMBAD.

Source 2, has three identifications in NED but none in SIMBAD. Two of the NED identification are galaxies the third is a point source. The first NED identification is the galaxy SDSS J090530.11-002654.5, and has apparent magnitude 18.7g. The second NED identification is the point source SDSS J090529.91-002649.9, and has apparent magnitude 17.0g. The third NED identification is the galaxy SDSS J090530.63-002656.3, and has apparent magnitude 21.2g.

Source 3, has one identification in NED but none in SIMBAD. The NED identification is a galaxy. The name of the galaxy is SDSS J090722.47+001835.9, and has apparent magnitude 18.9g.

Source 4, has one identification in NED but none in SIMBAD. The NED identification is a galaxy. The name of the galaxy is SDSS J090036.27+002527.1, and has apparent magnitude 19.7g.

Source 5, has no identification in NED or SIMBAD.

Source 6, has one identification in NED but none in SIMBAD. The NED identification is a galaxy. The name of the galaxy is SDSS J090722.28+002548.9, and has apparent magnitude 21.2g.

As laid out in section 4.3., the further source at greater than 5σ, in the PACS image but not seen in the SPIRE image, has two sets of right ascension and declination, very close to each other on the sky, for its location. They are referred to as A and B. No identifications are found in NED or SIMBAD for both sets of right ascension and declination.

The QSO stacking in outlined in section 4.4. The stacked 100μm flux is, 0.002987$\pm$9.34$\times 10^{-5}$Jy. The stacked 160μm flux is, 0.004762$\pm$0.001097Jy. The QSO catalogue was compared with the red PACS map to obtain the significance in the KS test at 160μm. The QSO catalogue was compared with the green PACS map to obtain the significance in the KS test at 100μm. The 7.43$\times 10^{-7}$% significance is seen in the KS test at 160μm, and 4.66$\times 10^{-6}$% significance is seen in the KS test at 100μm. These low values suggest we have a detection of star formation at both PACS red and green wave bands. See section 4.4. for details.

5.3 Further work

We need to deduce the SFRs of QSOs, using their redshifts. There are several star formation indicators in use today, for example: the Lyman alpha emission line (see section 1.1.2.1. for details); the ultraviolet continuum emission (see section 1.1.2.2 for details); the H alpha line emission (see section 1.1.2.3. for details); the far infrared continuum emission (see section 1.1.2.4. for details); the X-ray continuum emission (see section 1.1.2.5. for details) and the radio continuum emission (see section 1.1.2.6. for details). However in all these cases except far-IR luminosity, the central engine flux contaminates the signal from star formation. In all cases you need the luminosity of the QSO in the appropriate wavelength range and an assumed initial mass function to get the SFR. Hence you need to know its redshift. In the case of Blazars they also emit strongly in the far infrared, so in this case it is difficult to estimate SFRs.

QSO candidates can be detected by means of photo-z, colour selection and stellar morphology.

Spectroscopic redshifts are not always possible to obtain, especially if the QSO and host galaxy is very faint. When this happens photometric redshifts are obtained instead. This technique uses the broad colours of the QSO's host galaxy, to estimate the redshift. There are many filters that can be used to observe the QSO, this is essentially a spectrum with very poor wavelength resolution. In this case the broad shape of the spectrum is used to get the redshift estimate.

We need to deduce the SFRs of fainter QSOs. Quasars can be faint for two reasons: they have low luminosity and are relatively nearby, or they have high luminosity and are at high redshift. The SFRs of faint quasars is important because we need to understand the link between SFR and quasar luminosity. If the quasar is at high redshift the radiation from it came to us from earlier cosmic epochs, so this can help us understand the time dependence of star formation, by comparing it to lower z quasars of the same absolute magnitude.

The following studies could also be useful. We could fix z and see how the SFR depends on M_{abs}, absolute magnitude. We could also fix M_{abs} and see how the SFR depends on z. We can use this data to understand the SFR/QSO luminosity relation plus the SFR evolution with time.

Do the SEDs of QSOs, evolve with time? It is assumed in the SPIRE stacks, Serjeant et al. 2010, that they do not evolve with time. Calculating the stacked SPIRE and PACS colours will test this. Our sample of quasars is flux limited. If you plot a graph of M_{abs} against log(z) of the quasars in our sample, it will consist of three regions, one where the quasars are detected and the other two where they are not detected. The flux limit will be a roughly straight line with positive gradient. Above this line the quasars are detected, below this line they are not detected. The quasars will cluster close to the flux limit.

In the graph of the M_{abs} against z of the quasars in our sample, there will be a roughly triangular shaped region, lying above the region where the quasars are detected, that will be empty due to evolution of the QSO luminosity function. The third region is below the flux limit and is therefore also empty of quasars. Fainter QSOs will fill out the coverage of M_{abs}-z plane.

There is more than one flux limit in play with these quasars, because they come from several sources. It still remains true though that fainter quasars will improve the coverage of the M_{abs}-z plane.

We do not know whether all quasars are the same, and the various types that are seen, see section 1.2. for details, form an evolutionary sequence. Comparing the far-IR fluxes of different QSO types can constrain the evolutionary or orientational-based relationships between them. We also do not know if all galaxies were quasars in an early stage in their evolution. If these ideas are right, quasars could prove vital in understanding how galaxies form and evolve.

Adelberger K.L.,Steidel C.C.,Shapely A.E.,Pettini M.,2003,ApJ,584,45.

Alexian G.,Belkacem K.,Samadi R.,Valls-Gabaud D.,SF2A2011.pp299-302.

Antonucci S..1993.Unified models for active galactic nuclei and quasars.Annual Review of Astronomy and Astrophysics,31,473.

Balland C.,Devriendt J.E.,Silk J.,2003,MNRAS,343,107.

Balogh M. et al. 2004.MNRAS,348,1355.

Bamford S.P. et al.2009,Monthly Notices of the Royal Astronomical Society,393,1324.

Bartelmann M.2001.A&A,370,754.A&,388,732.

Baugh C.M.,Cole S.,Frenk C.S.,Lacey C.G.,1998,ApJ,498,504.

Baugh C.M.,Lacey C.G.,Frenk C.S.,Granato G.L.,Silva L.,Bressan A.,Benson A.J.,& Cole S.,2004,Mon. Not. R. Astron. Soc. 356,1191-1200.

Bawer R.G.,Benson A.J.,Malbon R.,Helly J.C.,Frenk C.S.,Baugh C.M.,Cole S., & Lacey C.G.,2006,MNRAS,370,645.

Bechtold J.,Yee H.K.C.,Elston R.,Ellingson E..1997.Ap.J.Lett.477:L29-L32.

Beelen A. et al. 2006.ApJ,642,694.

Bell E.F. et al.2004.Astrophysical Journal,608,752.

Bell E.F. et al.2006.The astrophysical journal,640:241-251.

Bender R.,Kormendy J. & Dehren W.1996.ApJ,464,L119.

Bendo G. et al. 2003.AJ,125,2361.

Benson A.J.,Lacey C.G.,Baugh C.M.,Cole S.,Frenk C.S.,2002b,MNRAS,333,156.

Benson A.J,Frenk C.S.,Baugh C.M.,Cole S., & Lacey C.G.,2003,MNRAS,343,679.

Benson A.J.,Bower R.G.,Frenk C.S.,Lacey C.G.,Baugh C.M.,Cole S.,2003,ApJ,599,38,B03.

Berlind A.A.,& Weinberg D.H.,2002,ApJ,575,587.

Biermann P..1976.Astron.Astrophys.53:295-303.

Blain A.W.,Smail I.,Ivison R.J.,Kneib J.P.,1999a,MNRAS,302,632.

Blain A.W.,Jameson A.,Smail I.,Longair M.S.,Kneib J.P.,Ivison R.J.,1999b,MNRAS,309,715.

Bregman J.N.,Snider B.A.,Grego L.,& Cox C.V.1998.ApJ,499,670.

Browne I.2003.MNRAS,341,13.

Buat V.,Deharveng J.M..1988.Astron.Astrophys.195:60-70.

Buat V.,Deharveng J.M.,Donas J., 1989. Astron. Astrophys. 223:42-46.

Buat V.,1992,Astron. Astrophys. 264:444-454.

Buat V.,Xu C..1996.Astron.Astrophys.306:61-72.

Bullock J.S.,Wechsler R.H.,& Somerville R.S.,2002,MNRAS,329,246.

Calzetti D.,Kinney A.L.,Storchi-Bergmann T..1994.Ap.J.429:582-601.

Calzetti D.,Kinney A.L.,Storchi-Bergmann T..1996.Ap.J.458:132-135.

Calzetti D..1997.Astron.J.113:162-184.

Caplan J.,Deharveng L..1986.Astron.Astrophys.155:297-313.

Caplan J.,Ye T.,Deharveng L.,Turtle A.J.,Kennicutt R.C..1996.Astron.Astrophys.307:403-416.

Chary R.,Elbaz D.,2001,ApJ,556,562.

Clark D.H.,Caswell J.L..1976.MNRAS.174:267-305.

Cohen J.G..1976.Ap.J.203:587-592.

Cole S.,Aragon-salamana A.,Frenk C.S.,Navarro J.F. & Zepf S.E.,1994,MNRAS,271,781.

Cole S.,Lacey C.G.,Baugh C.M. & Frenk C.S.,2000,MNRAS,319,168.

Collins C.A.,2009,Nature,458,603.

Condon J.J.,Yin Q.F..1990.Ap.J.357:97-104.

Condon J.J.,Anderson M.L.,Helou G..1991a.Ap.J.376:95-103.

Condon J.J..1992.Annu.Rev.Astron.Astrophys.30:575-611.

Conroy C.,Wechsler R.H.,& Kravtsov A.V.,2006,ApJ,647,201.

Conroy C.,Wechsler R.H.,2009,ApJ,696,620.

Cowie L.L.,Songaila A.,Hu E.M.,& Cohen J.G.,1996,AJ,112,839(839).

Cowie L.L.,Hu E.M.,Songaila A.,Egami E.,1997.Ap.J.Lett.481:L9-L13.

Cox P.,Mezger P.G..1989.Astron.Astrophys.Rev.1:49-83.

Croton D.J.,et al.,2006,MNRAS,365,11.

Davies J. & Burstein D.Editors.1995.The Opacity Of Spiral Disks.

Deharveng J.M.,Joubert M.,Kunth D.,1986,in the First, IAP workshop: Star-Forming Dwarf Galaxies And Related Objects, edited by D.Kunth,T.X. Thuan and J. Tran Thanh Van, Editions Frontieres, p.431.

Deharveng J.M.,Sasseen T.P.,Buat V.,Bawyer S.,Lampton M.,Wu X.,1994,Astron. Astrophys. 289:715-728.

Devereux N.,and Young J.S.1990.ApJ,359,42.

Devereux N.A.,Young J.S..1990.Ap.J.Lett.350:L25-L28.

Devereux N.A.,Hameed S..1997.Astron.J.113:599-608.

Devlin M.J. et al. 2009.Nature.458,737.

Devriendt J.E.G.,Guiderdoni B.,2000,A&A,363,851.

de Zotti G.,Ricci R.,Mesa D.,Silva L.,Mazzotta P.,Toffolatti L. & Gonzalez-Nuevo J.2005.A&,431,893.

Dijkstra M.,and Westra E.,Mon. Not. R. Astron. Soc. 401,2343-2348,2010.

Dole H. et al. 2006.A&A,451,417.

Donas J.,Deharveng J.M.,1984,Astron. Astrophys. 140:325-333.

Donas J.,Deharveng J.M.,Laget M.,Millard B.,Huguenin D.,1987,Astron. Astrophys. 180:12-26.

Donas J.,Millard B.,Laget M.,1995,Astron. Astrophys. 303:661-672.

Driver S. et al. 2007.MNRAS,379,1022.

Driver S.,Popescu C.,Tuffs R.,Graham A.,Liske J.,& Baldry I.2008.ApJ,678,101.

Driver S. et al. 2009.Astronomy and Geophysics,50,12.

Drory N.,Bender R.,Feuler G.,Hopp U.,Maraston C.,Snigula J.,Hill G.J.,2004,ApJ,608,742.

Dunne L.,Eales S. & Edmunds M.2003.MNRAS,341,589.

Dwek E. et al. 1998.ApJ,508,106.

Dye S. et al. 2007b.MNRAS,375,725.

Eales S. et al. 2009.

Elbaz D. et al. 2009.A&.507,1359.

Elbaz D.2009,ASPC,408,315E.

Ellis R.S.1997.Annu.Rev.Astron.Astrophys.35:389-443.

Engelbracht C.W..1997.Infrared observations and stellar populations modeling of starburst galaxies.PhD thesis,Univ.Arizona,Tempe.

Evans I.N.,Koratbar A.P.,Storchi-Bergmann T.,Kirkpatrick H.,Heckman T.M.,Wilson A.S..1996.Ap.J.Suppl.105:93-127.

Fanelli M.N.,Marcum P.M.,Waller W.H.,Corrett R.H.,O'Connell R.W.,et al.,1998. In The Ultraviolet Universe at Low and High Redshift, ed, W.Waller, M. Fanelli, J.Hallis, A. Danks,AIP. Conf. 408. New York:Am. Inst. Phys. In press.

Feinstein C..1997.Ap.J.Suppl.112:29-47.

Ferguson A.M.N.,Wyse R.F.G.,Gallagher J.S.,Hunter D.A..1996.Astron.J.111:2265-2279.

Ferrarese L.,Meritt D..2000.The astrophysical journal,539:L9-L12.

Fixsen D.J.,Dwek E.,Mather J.C.,Bennett C.L.,& Shafer R.A.1998.ApJ,508,123.

Fontanot F.,De Lucia G.,Monaco P.,Somerville R.S.,& Santini P.,2009,MNRAS,397,1776.

Gallagher J.S.,Hunter D.A..1984.Annu.Rev.Astron.Astrophys.22:37-74.

Gallego J.,Zamerao J.,Aragon-Salamanca A.,Rego M..1995.Ap.J.Lett.445:L1-L4.

Gavozzi G.,Boselli A.,Kennicutt R..1991.Astron.J.101:1207-1230.

Gerke F.,Newman J.A.,Davis M.,Coil A.L.,Cooper M.C.,Datton A.A.,Caber S.M.,Guhathakurtra P.,Konidaris N.,Koo D.C.,Lin L,Noeske K.,Philips A.C.,Rosario D.J.,Weiner B.J.,Willmer C.M.A.,Yan R..The Astrophysical Journal, Voulume 751.,Issue 1, article id.50.(2012).

Gilmore G.F.,Parry I.R.,Howell D.J.,eds.1998.The Stellar Initial Mass Function. San Francisco:Astron.Soc.Pac.In press.

Glazebrook K..2009.Nature.460,694.

Goldader J.D.,Joseph R.D.,Doyan R.,Sanders D.B..1995.Ap.J.444:97-112.

Goldader J.D.,Joseph R.D.,Doyan R.,Sanders D.B..1997.Ap.J.Suppl.108:449-470.

González Delgado R.M.,Perez E.,Tadhunter C.,Vilchez J.,Rodriguez-Espinoza J.M..1997.Ap.J.Supp 198.

Granato G.L.,Lacey C.G.,Silva L.,Bressan A.,Baugh C.M.,Cole S.,Frenk C.S.,2000,ApJ,542,710,G00

Granato G.L.,De Zotti G.,Silva L.,Bressan A.,Danese L.,2004,ApJ,600,580.

Griffin M.J. et al. Astronomy and Astrophysics manuscript no. Griffin 14159 astroph.

Grimm H.J.,Gilfanov M.,and Sunyaev R..2003.Mon.Not.R.Astron.Soc.339:793-809.

Guiderdoni B.,Hivon E.,Bouchet F.R.,Maffei B.,1998,MNRAS,295,877.

Guo Q.,White S.,Li C.,& Boylan-Kolchin M.,2010,MNRAS,404,1111.

Halland W.S. et al. 1998.SPIE.vol 3357:page 305.

Hartmann L.W.,Huchra J.P.,Geller M.J.,1984,ApJ,287,487.

Hartmann L.W.,Huchra J.P.,Geller M.J.,O'Brien P.,Wilson R.,1988,ApJ,326,101.

Hayes M.,Doctoral Dissertation 2007.Stockholm Observatory.Department of Astronomy.SE-

10691 Stockholm.

Heckman T.M.,Lehrert M.D.,Strickland D.K.,Armus L.,2000,ApJs,129,493.

Helou G.,Khan I.R.,Malek L.,Boebmer L..1988.Ap.J.Suppl.68:151-172.

Ho P.T.P.,Beck S.C.,Turner J.L..1990.Ap.J.349:57-66.

Hodge P.W.,Kennicutt R.C..1983.Astron.J.88:296-328.

Holmberg E.1958.Medd.Lund.Astron.Obs.Ser.2,6.

Hunter D.A.,Gallagher J.S.,1985.Ap.J.Suppl.58:533-560.

Hunter D.A.,Gillet F.C.,Gallagher J.S.,Rice W.L.,Low F,J..1986.Ap.J.303:171-185.

Hunter D.A.,Hawley W.N.,Gallagher J.S..1993.Astron.J.106:1797-1811.

Israel F.P.,van der Hulst J.M..1983.Astron.J.88:1736-1748.

Jing Y.P.,Mo H.J.,& Boerner G.,1998,ApJ,4941.

Kashlinsky A. & Atrio-Barandela F.2000.ApJ,536,L67.

Kaufman M.,Bash W.N.,Kennicutt R.C.,Hodge P.W..1987.Ap.J.319:61-75.

Kennicutt R.C.,Kent S.M..1983.Astron.J.88:1094-1107.

Kennicutt R.C..1983a.Ap.J.272:54-67.

Kennicutt R.C..1992b.Ap.J.Suppl.79:255-284.

Kennicutt R.C.,Tamblyn P.,Congdon C.W..1994.Ap.J.435:22-36.

Kennicutt R.C.,Bresolin F.,Bormans D.J.,Bothun G.D.,Thompson I.B..1995.Astron.J.109:594-604.

Kennicutt R.C.,Jr.,1998,ARA &,36,189-231.

Kinney A.L.,Bohlin R.C.,Calzetti D.,Panagia N.,Wyse R.F.G.,1993,Ap.J.Suppl.86:5-93.

Klien U.,Growe R..1986.Astron.Astrophys.161:155-168.

Kormendy J.1993a,In The Nearest Active Galaxies,ed.J.E.Beckman,L.Colina & H.Netzer,Madrid:CS

Kormendy J. & Richstone D.1995.ARA & A,33,581,KR95.

Kormendy J. et al. 1997.ApJ,482,L139.

Kunth D.,Terlevich E., Terlevich R.,Tenrio-Tagle G.,astro-ph,19809096. 8 Sep 1998.

Kunth D.,Mass-Hese J.M.,Terlevich E.,Lequeux J.,and Fall S.M.,Astron.,Astrophys.,334,11-20,1998.

Lamarre J-M.;Puget J.L.;Piat M.;Ade P.A.R.;Lange A.E.;Benoit A.;DeBernardis P.;Bauchet F.R.;Bock J.J.;Desert F.X.;Emery R.J.;Giard M.;Maffei B.;Murphy J.A.;Torre J-P.;Bhatia R.;Sudivala R.V.;Yourchemko V..IR Space Telescopes and Instruments.Edited by John C.Mather.Proceedings of the SPIE,Volume 4850.pp.730-739(2003).2003SPIE.4850.730L.

Larson R.B.,Tinsley B.M.,1978,ApJ,219:46-59.

Le Floch et al. 2005.ApJ.632:169.

Lehnert M.D.,Heckman T.M..1996.Ap.J.472:546-563.

Leitherer C.,Heckman T.M..1995.Ap.J.Suppl.96:9-38.

Leitherer C.,Ferguson H.C.,Heckman T.M.,Lowenthal J.D..1995a.Ap.J.Lett.454:L19-L22.

Leitherer C.,Robert C.,Heckman T.M.,1995b. Ap.J.Suppl.99:173-187.

Lonsdale C.J.,Helou G..1987.Ap.J.314:513-524.

Madau P., Ferguson H.,Dickinson M.,Giavalisco M.,Steidel C.C.,Fruchter A.,1996.MNRAS,283:1388-1404.

Madau P.,Pozzetti L.,Dickinson M.,1998.Ap.J.In press.

Maedr A..1987.Astron.Astrophys.173:247-262.

Magorrian J. et al. 1998.The astronomical journal,115:2285-2305.

Martin C.L..1998.Ap.J.In press.

Martin C.L.,1999,ApJ,513,156.

Massey P..1998.In The Stellar Initial Mass Function ed.B.F.Gilmore,I.R.Parry,D.J.Howell.San Francisco:Astron.Soc.Pac.In press.

Maz D.,Filippenko A.V.,Ho L.C.,Macchetto D.,Ric H-W.,Schneider D.P.,1996,Ap. J. Suppl. 107:215-226.

McIntosh D.H. et al.2005.The astrophysical journal,632:191-209.

Meier D.L.,Terlevich R.,1981,ApJ,246,L109.

Meurer G.R.,Heckmann T.M.,Leitherer C.,Kinney A.,Robert C.,Garnett D.R.,1995,Astron. J.110:2665-2691.

Meurer G.R.,Gerhardt R.,Heckman T.M.,Lehnert M.D.,Leitherer C.,Lauenthal J..1997.Astron.J.114 68.

Miller G.E.,Scalo J.M..1979.Ap.J.Suppl.41:513-547.

Mivilles-Deschenes M.A. et al. 2007.

Moshir M.,Kopan G.,Conrow J.,McCallon H.,Hacking P.,et al.1992.Explanatory Supplement to the IRAS Faint Source Survey, Version2,JPL:D-10015 8192.Jet Prop. Lab.,Pasadena,Calif.

Moster B.P.,Somerville R.S.,Maulbetsch C.,van den Bosch F.C.,Maccio' A.V.,Naab T.,& Oser L.,2010,ApJ,710,903.

Negrello M. et al. 2010.Science, volume 330,Issue 6605,pp.800-(2010).

Neistein E.,Li C.,Khochfar S.,Weinmann S.M., Shankar F.,& Boylan-Kolchin M.,2011b,MNRAS,sub arXiv:1103,3272.

Niklas S.,Klein U.,Brouine J.,Wielebinski R..1995.Astron.Astrophys.Suppl.114:21-49.

Niklas S.,Klein U.,Wielebinski R..1997.Astron.Astrophys.322:19-28.

Nortan A.Book.Introduction To Astrophysics.The Open University, Milton Keynes UK,2002.

Oey M.S.,Kennicutt R.C..1997.MNRAS.In press.

O'Mearea J.M.,Tyltler D.,Kirkman D.,Suzuki N.,Prochaska J.X.,Lubin D.,Wolfe A.M.,2001,ApJ,552

Patel K.,Wilson C.D..1995a.Ap.J.451:607-615.

Patel K.,Wilson C.D..1995b.Ap.J.453:162-172.

Peacock J.A.,& Smith R.E.,2000,MNRAS,318,1144.

Pearson C.,Rowan-Robinson M.,1996,MNRAS,283,174.

Pettini M.,Rix S.A.,Steidel C.C.,Adelberger K.L.,Hunt M.P.,Shapley A.E.,2002,ApJ,569,742.

Phillips A.C..1993.Star formation in barred spiral galaxies.PhD thesis.Univ.Washington,Seattle.

Pilbratt Göran. JENAM 2010, Joint European and National Astronomy Meeting held 6-10 September, 2010, in Lisbon Portugal. p 149.

Poglitsch A. et al. Astronomy and Astrophysics manuscript no. 14535 PACS. May 21, 2010.

Pozzetti L. et al.,2003,The K20 Survey,A&,402,637.

Priddley R. et al. 2003.MNRAS,339,1183.

Prouton O.R.,Bressan A.,Clements M.,Franceschini A.,Granato G.L.,Silva L.,2004,A&A,421,115.

Puxley P.J.,Brand P.W.J.L.,Moore T.J.T.,Mountain C.M.,Nakai N.,Yamashita A.T..1989.Ap.J.345: 168.

Puxley P.J.,Hawarden T.G.,Mountain C.M..1990.Ap.J.364:77-86.

Rieke G.H.,Lebofsky M.J.,Thompson R.I.,Low F.J.,Tokunaga A.T..1980.Ap.J.238:24-40.

Rodriguez-Puebla A.,Avila-Reeve V.,Firmani C.,& Colin P.,2011,Rev. Mex. AA,submitted,arXiv:11

Romanishin W..1990.Astron.J.100:373-376.

Rowan-Robinson M.,Crawford J..1989.MNRAS.238:523-558.

Ryder S.D.,Dopita M.A..1993.Ap.J.Suppl.88:415-421.

Ryder S.D.,Dopita M.A..1994.Ap.J.430:142-162.

Sage L.J.,Solomon P.M..1989.Ap.J.Lett.342:L15-L18.

Salpeter E.E.,1955,Ap.J.121:161-167.

Sauvage M.,Thuan T.X..1992.Ap.J.Lett.396:L69-L73.

Sauvage M.,Thuan T.X..1994.Ap.J.429:153-171.

Scaccimarro R.,Sheth R.K.,Hui L.,& Jain B.,2001,ApJ,546,20.

Scalo J.M..1986.Fundam.cosm.Phys.11:1-278.

Schawinski K.et al.2006,Nature,442,888.

Seljak U.,2000,MNRAS,318,203.

Seo H.,Eisenstein D.J.,& Zehavi I.,2008,ApJ,681,998.

Serjeant S. et al. 2003.Mon.Not.R.Astron.Soc.344,887-904.

Serjeant S. et al. 2010.A&,518,L7,2010.

Serjeant S. Book. Observational Cosmology.Cambridge University Press 2010.

Silva L.,Granato G.L,Bressan A.,Danese L.,1998,ApJ,509,103,S98.

Smail I.,Chapman S.C.,Ivison R.J.,Blain A.W.,Takata T.,Heckman T.M.,Dunlop J.S.,Sekiguchi K.,2003,MNRAS,342,1185.

Smith A.M.,Corrett R.H.,1982,Ap.J.261:1-11.

Smith B.J.,Harvey P.M.,1996,Ap. J. 468:139-166.

Somerville R.S.,Primack J.R.,Faber S.M.,2001,MNRAS,320,504.

Spergel D.N. et al.,2003,ApJs,148,175.

Steidel C.C.,Giavalisco M.,Pettini M.,Dickson M.,Adelberger K.L.,1996,ApJ,Lett.462:L17-L21.

Terlevich E.,Diaz A.I.,Terlevich R.,Garcia Vargas M.L.,1993,MNRAS,260,3.

Turner J.L.,Ho P.T.P..1994.Ap.J.421:122-139.

Ulvestad J.S..1982.Ap.J.259:96-102.

van der Hulst J.M.,Kennicutt R.C.,Crane P.C.,Rots.A.H..1988.Astron.Astrophys.195:38-52.

van der Marel R.P.,de Zeeuw P.T.,Rix H.W. & Quinlan G.D.1997b,Nature,385,610.

Vӧlk H.J.,Klein U.,Wielebinski R..1989.Astron.Astrophys.213:L12-L14.

Vio R.,Tenorio L. and Wamteker W.2002.A&A,391,789-794.

Wake D.A. et al.,2008,MNRAS,387,1045.

Wake D.A. et al.,2011,ApJ,728,46.

Waller W.,Fanelli M.,Danks A.,Hallis J.,eds. 1997. The Ultraviolet Universe at Low and High Redshifts,AIP Conf.408.New York:Am. Inst. Phys. In press.

Walterbos R.A.M.,Braun R..1994.Ap.J.431:156-171.

Walterbos R.A.M.,Greenawalt B..1996.Ap.J.460:696-710.

White M.,Zheng Z.,Brown M.J.I.,Dey A.,& Jannuzi B.T.,2007,ApJ,655,L69.

Xu C..1990.Ap.J.Lett.365:L47-L50.

Yang X.H.,Mo H.J.,& van den Bosch F.C.,2003,MNRAS,339,1057.

Young J.S.,Allen L.,Kenny J.D.P.,Lesser A.,Round B..1996.Astron.J.112:1903-1927.

Zehavi I.,et al.,2005,ApJ,630,1.

Zehavi I.,et al.,2011,ApJ,submitted,arXiv:1005,2413.

Zehavi I.,Pattri S.,& Zheng Z.,2011,arXiv:1104,0389v1.

Zemcov M. et al. 2007.MNRAS,376,1073.

Zheng Z.,Coil A.L.,& Zehavi I.,2007,ApJ,667,760,ZCZ07.

Zheng Z.,Zehavi I.,Eisenstein D.J.,Weinberg D.H.,& Jing Y.,2008,ApJ,707,554.

Appendix A

Details of subroutines

A.1 Final PACS pipeline

```
;pro pacs_pipeline_post_deglitch_v4


;-----------------------------------------------------------------------
;
; Set up crucial parameters for the data analysis
;
;-----------------------------------------------------------------------


;;; flash_threshold is my hard-wired kludge for marking the positions
;;; in the red array of the calibrator flashes, using the data stream
;;; from pixel zero,zero.

;;; mask_flashes is a byte flag: set to 1B to mask the flashes and 0B
;;; to opt out of masking the flashes.

;;; format is a string to define the format of the input file. I
;;; originally found that FITS files weren't being read in correctly,
;;; but an up-to-date version of ASTROLIB fixed that, so there's no
;;; longer any need to split up the FITS files on Achilles and create
;;; separate IDL .dat files.

;;; pix_size is the pixel size of the final image, in arcseconds.
;;; width is the beamswitching distance, in readouts. Note that the

;;; slew speed is six arcseconds between each readout for the ATLAS

;;; red array data, and twelve arcseconds between each readout for the

;;; green array data.

;;;the data set wanted (nicknamed for now green71, green70, red70, red71)
;;;can be selected using one of those nicknames

;data='green70'
data='green71'
;data='red70'
;data='red71'

;;;then the following case statement will do the rest
;;;note that the mean wavelength set is calculated straight from
;;;the corresponding filter response

CASE data of
   'red70'  : BEGIN
              ;data_directory = '/padata/gamma/users/sserjeant/1342187170_red/'
              data_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/'
                save_file_suffix = 'level1_1342187170_red_hipe_dg'
                imagename='level1_1342187170_red_deglitched4_calfit1.fits'
;               save_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/IDL_
                save_directory='/padata/gamma/users/dgoodfellow/'
;                flash_threshold = 20.5 ; appropriate for 1342187170_red
;                mask_flashes = 0B
                format='IDL'          ;; FORMAT FOR READING IN DATA
                pix_size = 3.0 ;; in arcseconds
                lam=166d-6
;;;;;;;;The following still need to be optimised
                fwhm_2d = 1.22*lam/3.5d / !dtor * 3600 / pix_size
                fwhm_1d = 4
        sigma = 2.10506
                width = 4
              END
   'red71'  : BEGIN
              ;data_directory = '/padata/gamma/users/sserjeant/1342187171red/'
              data_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/'
               save_file_suffix = 'level1_1342187171_red_hipe_dg'
               imagename='level1_1342187171red_deglitched2_calfit1.fits'
```

```
;              save_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/IDL_c
               save_directory='/padata/gamma/users/dgoodfellow/'
               format='IDL' ;; FORMAT FOR READING IN DATA
;               flash_threshold = 17.5 ; appropriate for 1342187171red
;               mask_flashes = 1B
               pix_size = 3.0 ;; in arcseconds
       lam=166d-6
;;;;;;;;The following still need to be optimised
                fwhm_2d = 1.22*lam/3.5d / !dtor * 3600 / pix_size
                fwhm_1d = 4
        sigma = 2.10506
                width = 4
              END
   'green70' : BEGIN
               ;data_directory = '/padata/gamma/users/sserjeant/1342187170blue/'
               data_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/'
               save_file_suffix = 'level1_1342187170green_hipe_dg'
;              save_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/IDL_o
               save_directory='/padata/gamma/users/dgoodfellow/'
       imagename= 'level1_1342187170green_deglitched3_calfit1.fits'

               format='FITS' ;; FORMAT FOR READING IN DATA
;               flash_threshold = 0
;               mask_flashes = 0B
               pix_size = 2.0 ;; in arcseconds
               lam=102d-6
;;;;;;;;The following still need to be optimised (*2 for green??Th)
                fwhm_2d = 1.22*lam/3.5d / !dtor * 3600 / pix_size
                fwhm_1d = 4
        sigma = 1.03744
                width = 4
              END
   'green71' : BEGIN
;               data_directory = '/padata/gamma/users/sserjeant/1342187171blue/'
               data_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/'
               save_file_suffix = 'level1_1342187171green_hipe_dg'
       imagename= 'level1_1342187171green_deglitched2_calfit1.fits'
;               save_file_suffix = 'green71'
;               save_directory = '/padata/gamma/users/rhopwood/hipe_deglitched_data/IDL_
                save_directory='/padata/gamma/users/dgoodfellow/'
                format='FITS' ;; FORMAT FOR READING IN DATA
;                flash_threshold = 0
;                mask_flashes = 0B
                pix_size = 2.0 ;; in arcseconds
;                readcol,'~/hatlas/filters/PACS-100.data',lam,res,/silent
;                lam=tsum(res*lam/tsum(res))/1d6
        lam=102d-6
                ;102.457
;;;;;;;;The following still need to be optimised (*2 for green??)
                fwhm_2d = 1.22*lam/3.5d / !dtor * 3600 / pix_size
                fwhm_1d = 4
        sigma = 1.03744
                width = 4
            END

ENDCASE

filename=data+'s'+strtrim(string(sigma),2)+'_'+'w'+ \$
             strtrim(string(width),2)+'_'+'fwhm'+ \$
                         strmid(strtrim(string(fwhm_1d),2),0,3)

print,'Data directory: ',data_directory

;-------------------------------------------------------------------
;
; Create a save directory if it doesn't already exist
;
;-------------------------------------------------------------------


pushd, save_directory
spawn,'ls | fgrep saves2',jnk
ok = where(jnk eq 'saves2', nok)
if(nok eq 0) then begin
    print,'Creating save directory.'
    spawn,'mkdir saves2',jnk
endif
popd

save_directory=save_directory+'saves2/'

print,'Save directory: ',save_directory

;The data saved in the saves1 directory, has sigma set to four in all cases. Sigma is th
;filtering. The data saved in the saves2 directory, has sigma set to 2.10506 for red70 a
;set to 1.03744 for green70 and green71.
;-------------------------------------------------------------------
;
; Next read in the data
;
```

```
;;THE GAMMA 15 DATA SEEMS TO HAVE AN EXTRA FITS EXTENTION
;;WHICH MAKES RA IMAGE18 AND DEC IMAGE19
;
;-----------------------------------------------------------------------

print,'Reading in the data...'

;;;Future data might have the ra and dec images at +1 ie image18 and image19....

;read_in_pacs_data, data_directory=data_directory, save_file_suffix=save_file_suffix, \$

;  image1,image2,image3,image4,image5,image6,image7,image8,image9,image10,\$
;      image11,image12,image13,image14,image15,image16,image17, \$
;          image18,image19,image20,format=format

if(keyword_set(debug)) then begin

   ; restore,/verbose, file='quicksave_for_debugging.dat' ;;; red71 test file
   ; restore,/verbose,file='quicksave_for_debugging1.dat' ;;; red70 test file
   ; restore,/verbose,file='quicksave_for_debugging2.dat' ;;; green70 test file
   ; restore,/verbose,file='quicksave_for_debugging3.dat' ;;; green71 test file
endif else begin
;;stop
    image1=mrdfits(data_directory+imagename,1,h1)
;image2=mrdfits(data_directory+imagename,2,h2)
    image17=mrdfits(data_directory+imagename,17,h17)
    image18=mrdfits(data_directory+imagename,18,h18)
endelse


;
;image1=image1[*,0:1,0:1]
;image17=image17[*,0:1,0:1]
;image18=image18[*,0:1,0:1]

;;;to print the real time image2 use FINETIME
;;;image2.finetime (but I dont know how to interpret it!!)
;;;image2.resetindex can be used to plot for time, but whats the point?

print,'...done reading in.'

;-----------------------------------------------------------------------
;
; Create a header and image for mapping the pixel data onto the sky
; OR: have already read one in.
;
```

```
;----------------------------------------------------------------------

;print,'Creating header and image for coadd...'

;create_pacs_skyimage, image17, image18, pix_size, skyimage, \$
;                                  skynoise, header_skyimage

;print,'...done.'

;----------------------------------------------------------------------
;
; Calculate the noise level in the data stream and create a mask
;
;----------------------------------------------------------------------

print,'Creating noise array...'

;;;now creating a mask to note the input NaN positions before they are
;;;replaced for the beam swithching.
;;also adds random noise at the flagged positions.

s = size(image1)
mask = dblarr(s[1],s[2],s[3])

create_pacs_noise_array_2, image1, noise1, dimage=dimage1, mask=mask, \$
                                          /fillmask

print,'...done.'

print,'Filling in the gaps...'

;;;now set the NaNs in image1 to noise at the median sky level
fill_in_the_gaps4, image1, mask

;x=indgen(s[1])
;for i=0,s[2]-1 do begin
;  for j=0,s[3]-1 do begin
;    bad=where(finite(image1[*,i,j]) eq 0,nbad)
;    if (nbad gt 0) then begin
;      mask[bad,i,j]=1B;bad data set to mask=1
;      ok=where(finite(image1[*,i,j]) eq 1,nok)
;      if (nok eq 0) then begin
;        image1[*,i,j]=0
;      endif else begin
;        image1[bad,i,j]=interpol(image1[ok,i,j],x[ok],x[bad])
;      endelse
;    endif
;  endfor
;endfor

print,'...done.'

;;;Calibration block residuals are dealt with during the HIPE de-glitching
;;;the mask has these positions flagged.
;if(keyword_set(mask_flashes)) then begin
;; simple minded and very conservative filter to get rid of calibrator flashes
;    iflash = where(image1[*,0,0] gt flash_threshold)
;    flashmask = bytarr(s[1])
;    for i=0,n_elements(iflash)-1 do flashmask[iflash[i]:iflash[i]+2000]=1B
;; set the regions where the calibrator flash is present to very high
;; noise levels
;    ok = where(flashmask eq 1B)
;    noise1[ok,*,*] = 1d10
;endif

;--------------------------------------------------------------------
;
; Make correction for 1/f noise
;
;--------------------------------------------------------------------

;print,'Making correction for 1/f noise using a noise-weighted beamswitching...'

;;;kernel for sky smoothing function now defined with initial case statement

;;;when testing out the parameters for the beamswitching the dimentions of the
;;;kernel may need to be altered accordingly. use KDIM.

;pacs_beam_switching, image1, noise1, sigma, width, \$
;                      filtered_image1,filtered_noise1, kdim=0

;print,'...done.'

;;; I haven't yet quite understood the need for this
;;; normalisation (now in pacs_beam_switching) though
;;; it's required by the form of the
;;; convolution iw_x_p/w_x_p2. This noise-weighted convolution was
;;; derived in the 1990s for submm point sources in an image with zero
;;; sky, but what we're doing here is estimating a background
;;; level in a data stream with a non-zero sky, and what I
;;; haven't yet figured out is a derivation for this
;;; normalisation based on the requirement of a minimum-variance
;;; estimator of this background that leaves the resulting timeline
;;; with a white noise power spectrum.

;normalisation = total(kernel^2)/total(kernel)

;; fix regions near calibrator flashes. My guess is that this
;; sky background estimate I'm using is OK when the noise
;; levels are uncorrelated with the signal and are distributed in time
;; with a white noise power spectrum, but this breaks down in
;; the region of the calibrator flash where there's a big
;; coherent region that's being masked out. Therefore once the
;; kernel touches the calibrator flash masked region, I have to ignore
;; the data, so I give all that data a very high noise value.

;if(keyword_set(mask_flashes)) then begin
;    print,'...then dealing with regions near calibrator flashes...'
;    flashes = where(flashmask eq 1B)
;    for i=0,n_elements(flashes)-1 do begin
;        imin = (flashes[i]-n_elements(kernel))>0
;        imax = flashes[i]
;        filtered_noise1[imin:imax,*,*] = 1d10
;        imin = flashes[i]
;        imax = (flashes[i]+n_elements(kernel))<(s[1]-1)
;        filtered_noise1[imin:imax,*,*] = 1d10
;    endfor
;endif

;;; finally use the smoothed time series data to make a sky
;;; subtraction

;print,'...then using the smoothed data to make a sky subtraction...'

;sky = 0*image1
;sky_noise = 0*noise1
;sky[width:s[1]-1,*,*] = \$
;  0.5*filtered_image1[0:s[1]-1-width,*,*]
;sky_noise[width:s[1]-1,*,*] = \$
;  sqrt(sky_noise[width:s[1]-1,*,*]^2 + \$
;       0.25*filtered_noise1[0:s[1]-1-width,*,*]^2)
;sky[0:s[1]-1-width,*,*] = sky[0:s[1]-1-width,*,*] + \$
;  0.5*filtered_image1[width:s[1]-1,*,*]
;sky_noise[0:s[1]-1-width,*,*] = \$
;  sqrt(sky_noise[0:s[1]-1-width,*,*]^2 + \$
;       0.25*filtered_noise1[width:s[1]-1,*,*]^2)

;;; close to the edges we can't get a good background
;;; measurement so we set the noise values of these regions to very
;;; high values.

;sky_noise[0:width-1,*,*] = 1d10
;sky_noise[s[1]-1-width:s[1]-1,*,*] = 1d10

;sky_noise[0:24,*,*] = 1d10
;sky_noise[s[1]-1-25:s[1]-1,*,*] = 1d10

;sky_subtracted_image1 = image1 - sky
;sky_subtracted_noise = sqrt(noise1^2 + sky_noise^2)

;sky_subtracted_image1[0:24,*,*] = 0
;sky_subtracted_image1[s[1]-1-24:s[1]-1,*,*] = 0

;print,'...all 1/f correction now done.'

;----------------------------------------------------------------------
;
; ;;Create a mask array
; Fill the mask
;
;-----------------------------------------------------------------------
;;mask is already created at the 1st make_noise_array section
;;mask = bytarr(s[1],s[2],s[3])
;bad = where(sky_subtracted_noise gt 1d5, nbad)
;if(nbad gt 0) then mask[bad]=1B

;----------------------------------------------------------------------
;
; Make kernel and timeline filter
;
;----------------------------------------------------------------------

filtered_image1=dblarr(s[1],s[2],s[3])
filtered_noise1=dblarr(s[1],s[2],s[3])
for i=0,s[2]-1 do begin
  printdone, i, s[2]
  for j=0,s[3]-1 do begin
    ;i=50
    ;j=12
    delvarx, kernel
    sserjeant_vio_timeline_filter5,image1[*,i,j],noise1[*,i,j],sigma,c,d, kernel=kernel
    ;stop
    filtered_image1[*,i,j]=c
    filtered_noise1[*,i,j]=d
    ;save, kernel, file=save_directory+'kernel_'+strcompress(i,/rem)+'_'+strcompress(j,/
  endfor
endfor
printdone, i, s[2],/newline
;stop

;--------------------------------------------------------------------
; Re-measure the noise. I'm not sure why this is necessary yet
; (perhaps a mis-coding of the normalisation of the w_x_p2 term, or
; perhaps taking out the 1/f component changes things in some way?).
;
; This may not be the case now, need to check...
;
;--------------------------------------------------------------------

print,'Remeasuring the noise...'

zero=where(abs(filtered_noise1) lt 1.0d-10,nzero)
if (nzero gt 0) then filtered_noise1[zero]=1.0d10
;create_pacs_noise_array,filtered_image1,filtered_noise1,dimage=dimage1_version2,mask=ma
;;;; NOT NECESSARY ON SMOOTHED DATA

print,'...done.'

;--------------------------------------------------------------------
;
; Save the files created so far
;
;--------------------------------------------------------------------


if(not keyword_set(debug)) then begin
    print, 'Saving timeline results...'
    save, noise1, \$
      file=save_directory+'noise_'+save_file_suffix+'.dat'
;    save, sky, \$
;      file=save_directory+'saves/sky_'+save_file_suffix+'.dat'
;    save, sky_noise, \$
;      file=save_directory+'saves/sky_noise_'+save_file_suffix+'.dat'
    save, filtered_image1, \$
      file=save_directory+'filtered_timeline_image_'+save_file_suffix+'.dat'
```

```
;    save, sky_subtracted_noise, \$
;      file=save_directory+'saves/sky_subtracted_timeline_noise_'+save_file_suffix+'.dat
    save, filtered_noise1, \$
      file=save_directory+'filtered_timeline_noise_'+save_file_suffix+'.dat'
    print,'...done.'
endif


;---------------------------------------------------------------------
;
; Project the timeline data onto the sky
;
;---------------------------------------------------------------------

;project_pacs_onto_sky, sky_subtracted_image1, image17, image18, noise1, \$
;                       skyimage, skynoise, header_skyimage, 2d, mask, \$
;                       x_pixel=x_pixel, y_pixel=y_pixel

print,'Projecting timeline data onto the sky...'

make_sky_master_image_for_pacs_raster_version2,image17,image18,pacs_raster_master_image,

image0=filtered_image1
;image0[*,*,*]=!dpi
noise=filtered_noise1
master_image=pacs_raster_master_image
si=size(master_image)
master_noise=dblarr(si[1],si[2])
master_header=header_pacs_raster_master_image

coadd_timelines_into_master_pacs_image_version2,image0,noise,image17,image18,master_imag

;project_pacs_data_onto_sky, skyimage, skynoise, header_skyimage, image17, image18, mask
;  sky_subtracted_image1, noise1, x_pixel=x_pixel, y_pixel=y_pixel

print,'...done projection.'

;stop
;---------------------------------------------------------------------
;
; Filter in cross-scan direction
;
;---------------------------------------------------------------------

fwhm_in_arcseconds = 1.22*lam/3.5d / !dtor * 3600
```

```
xyad,master_header,[0,0],[0,1],ra_pix,dec_pix
gcirc,1,ra_pix[0]/15,dec_pix[0], ra_pix[1]/15,dec_pix[1],master_pixel_scale
fwhm_in_master_pixels = fwhm_in_arcseconds / master_pixel_scale[0]
size_of_kernel = 4*fwhm_in_master_pixels + 1

make_kernel_for_cross_scan_filtering,size_of_kernel,fwhm_in_master_pixels,kernel

scuba_src_ext,master_image,master_noise,kernel,flux,deltaflux,/no_ft

;stop

;-----------------------------------------------------------------------
;
; Save results
;
;-----------------------------------------------------------------------

if(not keyword_set(debug)) then begin
    print, 'Saving zerofootprint results...'

    save,file=save_directory+'zerofootprint_signal_'+save_file_suffix+'.dat', \$
      master_image,master_header
    save,file=save_directory+'zerofootprint_noise_'+save_file_suffix+'.dat', \$
      master_noise,master_header

    snrsky = imagenoise2snr(master_image,master_noise)

    sky_stats,snrsky[where(snrsky ne 0)],mout,vout
    print,'SNR sigma ',sqrt(vout)

    save,file=save_directory+'zerofootprint_snr_'+save_file_suffix+'.dat', \$
      snrsky,master_header

    print,'...done.'
endif

;-----------------------------------------------------------------------
;
; 2D filtering
;
;-----------------------------------------------------------------------

;print, 'Performing 2D filtering...'

;;;
```

```
;;;if fwhm_2d is changed then npix may need changing too
;;;

;psf = psf_gaussian(fwhm=fwhm_2d,npix=[17,17])
;scuba_src_ext, skyimage, skynoise, psf, flux, deltaflux, /no_ft

;print,'...done.'

;---------------------------------------------------------------------
;
; Save results
;
;---------------------------------------------------------------------

if(not keyword_set(debug)) then begin
    print, 'Saving filtering results...'

    save,file=save_directory+'filtered_signal_'+save_file_suffix+'.dat', \$
      flux,master_header

    save,file=save_directory+'filtered_noise_'+save_file_suffix+'.dat', \$
      deltaflux,master_header

    print,'...done.'
endif

;---------------------------------------------------------------------
;
; Scale deltaflux
;
;---------------------------------------------------------------------


snr = imagenoise2snr(flux, deltaflux)

if(not keyword_set(debug)) then begin
    print, 'Saving filtering snr results...'

    save,file=save_directory+'filtered_snr_'+save_file_suffix+'.dat', \$
      snr,master_header

    scale_pacs_deltaflux, flux, deltaflux, snr, sig1, sig2 ;, /silent

    save,file=save_directory+'filtered_scaled_var1_snr_'+save_file_suffix+'.dat', \$
      snr,master_header
    print,'...done.'

endif

;---------------------------------------------------------------------
;
; Closing
;
;---------------------------------------------------------------------


print,'...done.'

end
```

A.2 Timeline filtering

```
pro sserjeant_vio_timeline_filter5,input_signal_plus_fluctuations,$
        noise_level,fwhm,output_timeline,output_noise,$
kernel=kernel
;
;input_signal_plus_fluctuations, is an array giving the input flux
;as a function of time.
;fluctuations, is an array giving the noise as a function of time.
;fwhm, is the full width at half maximum of the psf, point spread
;function, of the sources we are looking for,
;in pixels.
;output_timeline, is an array giving the filtered signal.
;noise1, is the array giving the modified noise, such that,snr,
;signal to noise array has unit variance.
;psf, is the point spread function generated in the program,
;it is also an array.
;snr, is the signal to noise array, generated in the program.
;kernel, is the kernel used for filtering generated in this program,
;it is also an array.
;
;1. Convert the,fwhm, to the standard deviation,theta , of the psf.
;
theta=fwhm/(2.0d0*sqrt(alog(4.0d0)))
;print,'theta=',theta
;
```

```
;2. Compute the power frequency spectrum,p, of the noise array.
;That is the input_signal_plus_fluctuations.
;f, is the Fourier transform of the input signal plus noise fluctuations.
;p, is the power frequency spectrum of the noise array.
;
f=fft(input_signal_plus_fluctuations,-1)
p_unshifted=f*conj(f)



;
;3. Find the best fit values for,d, and,gamma1, in, p=d*q^(-gamma1).
;Where,q, is the frequency or Fourier space
;variable.
;d, is the best fit value for the constant multiple term in the power
; law best fit of the power frequency spectrum.
;gamma1, is the best fit value for the exponent term in the power law
; best fit of the power frequency spectrum.
;xn, is the pixel value of the time in the noise array.
;lx, is the log to the base e of xn.
;lp, is the log to the base e of p.
;fitp, is the output array of values for,a, and,b,from ladfit.
;a, is the constant term in the straight line of best fit, from
; ladfit, for,lx, against,lp.
;b, is the gradient of the straight line of best fit, from ladfit,
; for,lx, against,lp.
;number_of_bins, is the number of bins used in fitting a power law
; to the power frequency spectrum.
;xnrange, is the minmax array of array xn.
;xnmin, is the left hand end of a bin in the binned power spectrum data.
;xnmax, is the right hand end of a bin in the binned power spectrum data.
;xn_bin_middle, is the position of the centre of a bin in the binned
; power spectrum data.
;p_binned, is the mean value of p of all the values in a bin in the
; binned power spectrum data.
;



xn=findgen(n_elements(p_unshifted))
p=shift(p_unshifted,n_elements(p_unshifted)/2+1)
```

```
xn=shift(xn,max(xn)/2+1)
ok=where(xn ge max(xn)/2)
xn[ok]=xn[ok]-max(xn)

;;if the number of elements in xn is even OR odd then
;;xn[n_elements(xn)/2]=0.0 exactly


if(2*(n_elements(xn)/2) eq n_elements(xn)) then begin ;;; EVEN CASE
p_average = dblarr(n_elements(xn)/2)
p_average[*] = 0.5*(p[n_elements(xn)/2:n_elements(xn)-1] + $
              reverse(p[0:n_elements(xn)/2-1]))
xn_average = xn[n_elements(xn)/2:n_elements(xn)-1]
endif else begin ;;; ODD CASE
p_average = dblarr(n_elements(xn)/2+1)
; p_average[n_elements(p_average)-1] = p[0]
        p_average[0]=p[0]
p_average[1:*] = 0.5*(p[n_elements(xn)/2+1:n_elements(xn)-1] + $
         reverse(p[1:n_elements(xn)/2]))
xn_average = [xn[n_elements(xn)/2+1:n_elements(xn)-1],abs(xn[0])]
endelse

smoothing_scale = 2000
pfit = exp( median(alog(p_average),smoothing_scale) )
q_for_pfit = xn_average*!dpi/max(xn_average)

half_smoothing_scale = nint(smoothing_scale/2.0)+1
;;; remove the median problem with the small-scale structure
n = n_elements(q_for_pfit)
bad = where(q_for_pfit gt q_for_pfit[n-1-half_smoothing_scale])
;ok = where(q_for_pfit le q_for_pfit[n-1-half_smoothing_scale])
;; don't do this extrapolation because
;pfit[bad] = interpol(pfit[ok],q_for_pfit[ok],q_for_pfit[bad])
;; extrapolates from last 2 points ONLY.
ok = where(q_for_pfit gt q_for_pfit[n-1-smoothing_scale] and $
   q_for_pfit le q_for_pfit[n-1-half_smoothing_scale])
result = ladfit(q_for_pfit[ok],pfit[ok])
pfit[bad] = result[1]*q_for_pfit[bad] + result[0]




;4. Compute the,psf, and kernel, in Fourier space.
;fft_of_kernel, is the kernel in Fourier space.
;q, is the frequency or Fourier space variable.
```

```
;psf, is the psf in ordinary space.
;kernel, Vio filter, fltarr(npix). If ,gamma1=0, this is equivalent to
;a psf with a peak of one.
;a1, is the value of a in the equation, tau(x)=exp(-a1*x^2). tau,
;is the symbol for the psf.
;psf, must have an odd number of elements.
;npix, is the number of elements in the input timeline plus fluctuations.
;x, is the real space or ordinary space variable.
;
a1=1.0d0/(2.0d0*theta^2)
;print,'a=',a1
npix=n_elements(input_signal_plus_fluctuations)
if (2*(npix/2) eq npix) then begin
  npix=npix-1
endif else begin
  npix=npix
endelse
;print,'npix=',npix
x=findgen(npix)-(npix/2)
psf=exp(-a1*x^2)
;Find fft of psf.
fft_of_psf=fft(shift(psf,n_elements(psf)/2+1))*npix
;Have to multiply by npix to make IDL FFT conform to Vio definition.
q=2.0d0*!dpi*findgen(n_elements(x))/n_elements(x)
high=where(q gt !dpi,nhigh)
if (nhigh ge 0) then q[high]=q[high]-2.0d0*!dpi

fft_of_kernel = fft_of_psf/interpol(sqrt(pfit),q_for_pfit,abs(q))
; then interpolate the smallest values
;
middle = where(q eq min(abs(q)))
fft_of_kernel[middle] = 0.0
bad = where(q gt min(abs(q)) and q le q_for_pfit[half_smoothing_scale])
ok = where(q gt q_for_pfit[half_smoothing_scale] or q eq min(abs(q)))
fft_of_kernel[bad] = interpol(fft_of_kernel[ok],q[ok],q[bad])
bad = where(q ge -1*q_for_pfit[half_smoothing_scale] and q lt min(abs(q)))
ok = where(q lt -1*q_for_pfit[half_smoothing_scale] or q eq min(abs(q)))
order = sort(q[ok])
fft_of_kernel[bad] = interpol(fft_of_kernel[ok[order]],q[ok[order]],q[bad])




;5. Perform the inverse Fourier transform, to obtain the, kernel.
;x, is the real space or ordinary space variable.
```

```
;
if(n_elements(kernel) eq 0) then begin
kernel_big=double(shift(fft(fft_of_kernel,/inverse),$
           n_elements(psf)/2))/npix
; kernel_big=kernel_big*sqrt(!dpi)
kernel_big = kernel_big / max(kernel_big)

;Restrict size of kernel to increase speed of timeline filtering.
kmax=where(kernel_big eq max(kernel_big))
kmax=kmax[0]
;delta, is the semi width of the reduced kernel in pixels,
;        centered on the kernel's peak.
delta=50
kernel=kernel_big[kmax-delta:kmax+delta]
delta250=250
kernel250 = kernel_big[kmax-delta250:kmax+delta250]
delta500=500
kernel500 = kernel_big[kmax-delta500:kmax+delta500]
endif
;


;;; trim off the LSS in the kernel

trim_kernel, kernel, output_kernel
kernel = output_kernel


;6.Compute the input flux of the input signal plus fluctuations.
;This is the area between it and the x axis.
;fluxin, is the input flux.
;
fluxin=total(input_signal_plus_fluctuations)
;print,'fluxin=',fluxin
;




;7.Compute the,snr, signal to noise array.
;
snr=input_signal_plus_fluctuations/noise_level
bad=where(noise_level eq 0,nbad)
if (nbad gt 0) then begin
  snr[bad]=0.0d0
```

```
endif
nan=where(finite(noise_level) eq 0,number_of_nans)
if (number_of_nans gt 0) then begin
  snr[nan]=0.0d0
endif
;



;8.Adjust the noise array to give modified noise array, noise1.
;So that using the modified noise in evaluating the,
;snr, array the,snr, has variance one. This is done by finding the
;multiplier to produce the modified noise array.
;
ok=where(snr ne 0,nok)
;print,'snr nok=',nok
;sky_stats,snr[ok],skymean,skyvar
;skystd=sqrt(skyvar)
snr_ok = snr[ok]
snr_ok_diff = snr_ok[1:n_elements(snr_ok)-1]-snr_ok[0:n_elements(snr_ok)-2]
;sky_stats,snr_ok_diff,skymean,skyvar
;skystd = sqrt(skyvar)/sqrt(2.0d0)
skystd = stdev_pclip(snr_ok_diff, skymean)/sqrt(2.0)
;; we do this in preference to sky_stats because of the
;;digitisation in the input data
;print,'skymean=',skymean
;print,'skyvar=',skyvar
;print,'skystd=',skystd
noise1=noise_level*skystd
;



;9.Compute the weighted noise array,w.
;
w=1.0d0/noise1^2
bad=where(noise1 eq 0,nbad)
if (nbad gt 0) then begin
  w[bad]=0.0d0
endif
nan=where(finite(noise1) eq 0,number_of_nans)
if (number_of_nans gt 0) then begin
  w[nan]=0.0d0
```

```
endif
;



;10.Multiply the, input_signal_plus_fluctuations, by the weighted
; noise,w, element by element to obtaine,sw.
;
sw=input_signal_plus_fluctuations*w
nan=where(finite(sw) eq 0,number_of_nans)
if (number_of_nans gt 0) then begin
  sw[nan]=0.0d0
endif
;



;11.Then convolve,sw, with the,kernel.
;sw_x_p, is the array obtained by convolving the,sw, array with the kernel.
;

;sw_x_p=convol(sw,kernel,center=1,/edge_wrap)
sw_x_p = convolve_with_mirrors(sw,kernel,center=1)

;; remove drifts because the kernel isn't perfect (perhaps because the
;; kernel constructed from the
;; power spectrum isn't perfect) - the median scale is the size of the
;; ENTIRE kernel
remove_long_drifts, sw_x_p, sw_x_p_nodrifts, width=2*delta
sw_x_p = sw_x_p_nodrifts


;stop


;
;12.Next convolve,w, with the,kernel, squared.
;w_x_p2, is the array obtained by convolving the,w, array with the
; kernel squared.

;w_x_p2=convol(w,kernel^2,center=1,/edge_wrap)
w_x_p2 = convolve_with_mirrors(w,kernel^2,center=1)
```

```
test=where(w eq 0,ntest)
;print,'The number of elements where w is zero ',ntest
test=where(finite(w) eq 0,ntest)
;print,'The number of elements where w is not finite ',ntest
;print,'minmax kernel ',minmax(kernel)
;



;13.Compute the output_timeline, by dividing the result in section 11,
; by the result in section 12, and then
; multiply by c the calibration factor.
;
;print,'c=',c
;c=1.0d0
;c = theta*sqrt(2*!dpi)
output_timeline=sw_x_p/w_x_p2
;output_timeline=c*output_timeline

bad=where(w_x_p2 eq 0,nbad)
if (nbad gt 0) then begin
  output_timeline[bad]=0.0d0
endif
nan=where(finite(w_x_p2) eq 0,number_of_nans)
if (number_of_nans gt 0) then begin
  output_timeline[nan]=0.0d0
endif

output_noise = 1.0d0/sqrt(w_x_p2)
if(nbad gt 0) then output_noise[bad] = 1d10
if(number_of_nans gt 0) then output_noise[nan] = 1d10

;;;
;;; Determine the flux calibration factor
;;;

psfmax = where(psf eq max(psf))
truncated_psf = psf[psfmax[0]-100L:psfmax[0]+100L]
caltest = convol(truncated_psf,kernel,/edge_wrap)
;;; Calculated PSF has maximum value of 1 and total flux of
;;; total(psf).
;;; We want the output to be the total flux seen by the pixel as it
;;; scans across the source.
;;; In the absence of 1/f, the kernel is just the PSF and a point
;;; source with a peak of 1.0 will come out with a reported peak
;;; flux of
;;; max(convol(truncated_psf,truncated_psf,/edge_wrap))/total(psf^2)
;;; This must be equal to 1.0 exactly regardless of the PSF.
;;; What we want it to report is the total flux in the PSF, so we
;;; need to multiply by total(psf).
;;; However in the case of 1/f noise, the kernel is different, so
;;; a point source with the shape of the PSF will have a reported
;;; flux of
;;; F = max(convol(truncated_psf,kernel,/edge_wrap))/total(kernel^2)
;;; We need to correct this so we need to divide by F and then
;;; multiply by total(psf) for the reasons given above.

F = max(caltest)/total(kernel^2)
calibration_factor = total(psf)/F

output_timeline = output_timeline * calibration_factor
output_noise = output_noise * calibration_factor

;;; Correct noise level
output_noise = output_noise * stdev(output_timeline/output_noise)

;
;14.Compute the output flux of the output_timeline. This is the area
; between it and x axis.
;fluxout, is the output flux.
;
fluxout=total(output_timeline)
;print,'fluxout=',fluxout
;print,'The number of elements where w_x_p2 is zero ',nbad
;print,'The number of elements where w_x_p2 is not finite ',number_of_nans
;stop
;
;
end
```

A.3 Cross-scan filtering

This is a description of the cross-scan filtering programmes. There are four subprogrammes that are strongly connected to and consistute the cross-scan filtering part of the third version of the PACS pipeline programme. The subprogrammes make_sky_master_image_for_pacs_raster_version2.pro and

coadd_timelines_into_master_pacs_image_version2.pro are strongly connected to the cross-scan filtering. They are needed to project the timeline data onto the sky which is the input form needed for the cross-scan filtering routines. The subprogrammes make_kernel_for_cross_scan_filtering.pro and scuba_src_ext.pro consistute the cross-scan filtering. The subprogramme scuba_src_ext.pro does the cross-scan filtering and is an in house IDL subprogramme. There are two other in house IDL subprogrammes that generate inputs needed for the cross-scan filtering after timeline data has been projected onto the sky. These are xyad and gcirc which were discussed in section 3.2.1. The subprogrammes make_sky_master_image_for_pacs_raster_version2.pro, coadd_timelines_into_master_pacs_image_version2.pro and make_kernel_for_cross_scan_filtering.pro were written by myself together with help from my supervisor.

Inside the subprogramme make_sky_master_image_for_pacs_raster_version2.pro, the inputs to the subprogramme are image17, image18 and the optional input debug. The array image17 contains the right ascension for each PACS pixel as a function of time. The array image18 contains the declination for each PACS pixel as a function of time. Inside the subprogramme make_sky_master_image_for_pacs_raster_version2.pro, the outputs from the subprogramme are the blank array pacs_raster_master_image and the header header_pacs_raster_master_image. These outputs from the subprogramme make_sky_master_image_for_pacs_raster_version2.pro are used to generate the inputs needed for the subprogramme coadd_timelines_into_master_pacs_image_version2.pro. This is outlined in section 9 of the third version of the PACS pipeline programme in section 3.2.1.

The variable ra_start is the right ascension of the starting point of the PACS data set, which is equal to the array element image17[0,0,0]. The variable ra_end is the right ascension of the end point, used for evaluating the scan angle, of the PACS data set, which is equal to the array element image17[999,0,0]. The variable dec_start is the declination of the starting point of the PACS data set, which is equal to the array element image18[0,0,0]. The variable dec_end is the declination of the end point, used for evaluating the scan angle, of the PACS data set, which is equal to image18[999,0,0]. These two pixels chosen for the evaluation of the scan angle are used because they are close enough to each other to use the small great circle distance approximation and are far enough away from each other to be not seriously effected by dithering. Dithering occurs when a new bright star comes in to the field of view of the star tracker, device designed to keep pointing in the right direction, so the tracking system thinks that the telescope is now pointing in the wrong direction. So it moves the telescope then it soon realizes that it has made a mistake and comes back again.

The variable delta_dec is equal to the variable dec_end minus the variable dec_start. The variable delta_ra is given below,

$$delta\_ra = (ra\_end - ra\_start) * cos(!dtor * 0.5 * (dec\_start + dec\_end)). \qquad (A.1)$$

The IDL constant !dtor is the conversion factor for converting degrees to radians and

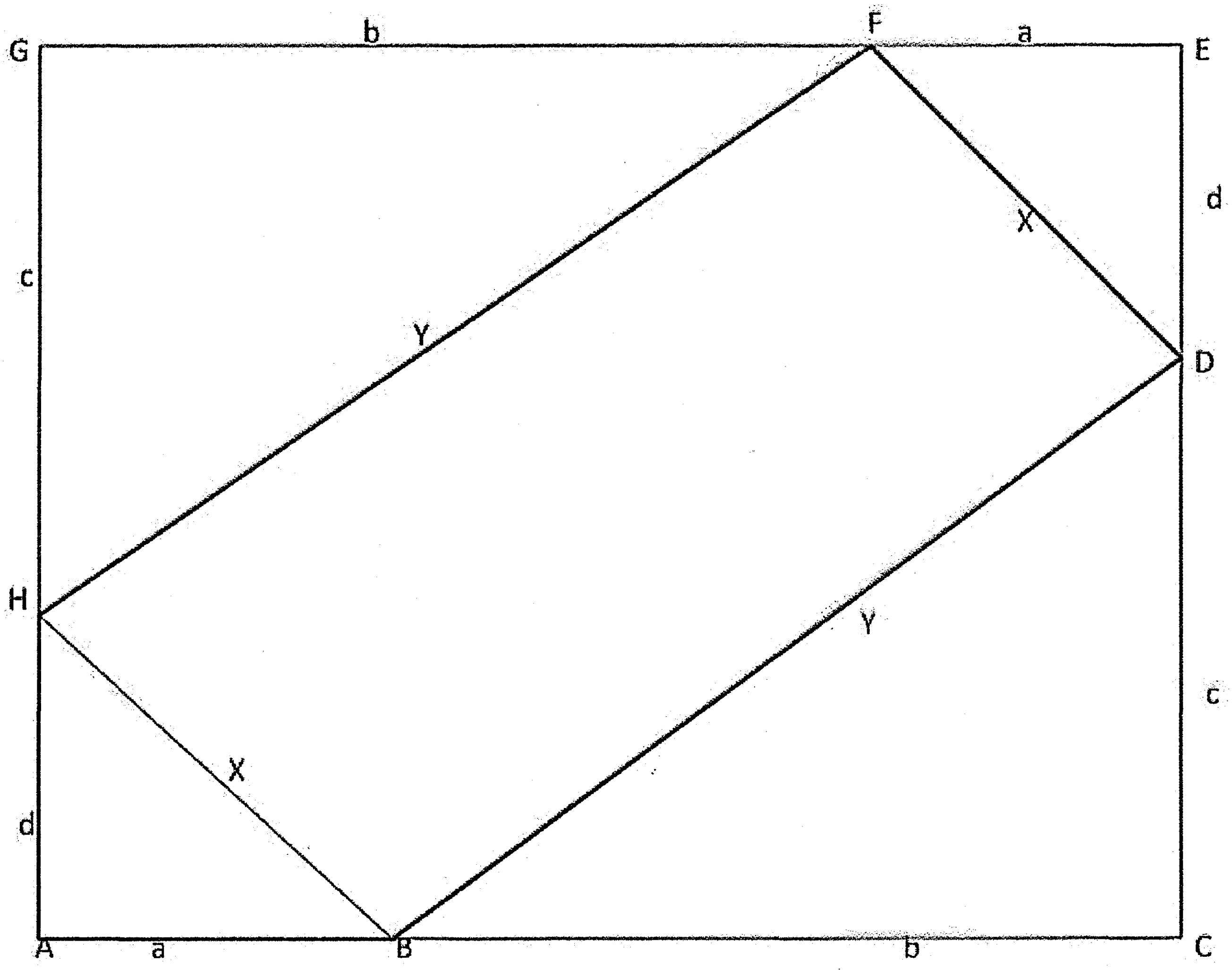


Figure A.1: This is the diagram of the starting positions used in cross scanning a PACS image.

has the value of $\frac{\pi}{180}$. The variable scan_angle is given below,

$$scan\_angle = \frac{atan(abs(delta\_dec), abs(delta\_ra))}{!dtor}. \tag{A.2}$$

abs is the built IDL function that gives the absolute value of the input.

Using the above diagram for reference, there are eight ways to scan the sky. There are four starting points and two directions to go in to begin the scan. I will now list them.

1. Start at B and begin by moving along BD.

2. Start at B and begin by moving along BH.
3. Start at D and begin by moving along DB.
4. Start at D and begin by moving along DF.
5. Start at F and begin by moving along FH.
6. Start at F and begin by moving along FD.
7. Start at H and begin by moving along HF.
8. Start at H and begin by moving along HB.

The variable ramin is the minimum value of the right ascension in the PACS data set. The variable ramax is the maximum value of the right ascension in the PACS data set. The variable decmin is the minimum value of the declination in the PACS data set. The variable decmax is the maximum value of the declination in the PACS data set. The variable typical_dec is equal to the mean value of the variables decmin and decmax.

The location of where the declination in the PACS data set is the minimum is found. The location of where the declination in the PACS data set is the maximum is found. The location of where the right ascension in the PACS data set is the minimum is found. The location of where the right ascension in the PACS data set is the maximum is found. The variable rawdecmin is the right ascension where the declination is the minimum for the PACS data set. The variable rawdecmax is the right ascension where the declination is the maximum for the PACS data set. The variable decwramin is the declination where the right ascension is the minimum for the PACS data set. The variable decwramax is the declination where the right ascension is the maximum for the PACS data set.

Using the diagram in this section for reference the following variables are defined.

The variables a, b, c, a, x and y are lengths in decimal degrees. The variable a is defined below,

$$a = (rawdecmin - ramin) * cos(typical\_dec*!dtor). \tag{A.3}$$

The variable b is defined below,

$$b = (ramax - rawdecmin) * cos(typical\_dec*!dtor). \tag{A.4}$$

The variable c is equal to the variable decwramax minus the variable decmin. The variable d is equal to the variable decmax minus the variable decwramax. The variable x is defined below,

$$x = sqrt(d^2 + a^2). \tag{A.5}$$

The variable y is defined below,

$$y = sqrt(b^2 + c^2). \tag{A.6}$$

It is important to note that we have already defined ra_start, dec_start, ra_end and dec_end for the first scan of this raster. However, ra_end is not the end of the scan but rather some intermediary point. We can make a rough estimate of the positions of the raster corners labelled B, D, F and H in the diagram in this section.

The variable ra_B is defined below,

$$ra\_B = ramin + \frac{a}{cos(typical\_dec*!dtor)}. \tag{A.7}$$

The variable dec_B is equal to the variable decmin. The variable ra_D is equal to the variable ramax. The variable dec_D is equal to the variable decmin plus the variable c. The variable ra_F is defined below,

$$ra\_F = ramin + \frac{b}{cos(typical\_dec*!dtor)}. \tag{A.8}$$

The variable dec_F is equal to the variable decmax. The variable ra_H is equal to the variable ramin. The variable dec_H is equal to the variable decmin plus the variable *d*.

gcirc the built in IDL subprogramme is used to determine which point out of B, D, F and H is closest to the starting point specified by ra_start and dec_start. Given the right ascension and declination of a pair of points on the sky gcirc gives you great circle distance between them. A similar thing can be done for a collection of points. This allows to define the variable closest. The variable closest can take one of four values 0, 1, 2 and 3. If closest is equal to zero then we start at point B. If closest is equal to one then we start at point D. If closest is equal to two then we start at point F. If closest is equal to three then we start at H.

In this subprogramme we have assumed that we are not straddling the right ascension of zero hours. If the variable closest is equal to zero and if ra_end is greater than ra_start, then we are starting at point B and are going right. If the variable closest is equal to zero and if ra_end is less than ra_start, then we are starting at point B and are going left. If the variable closest is equal to one and if dec_end is less than dec_start, then we are starting at point D and are going down. If the variable closest is equal to one and if dec_end is greater than dec_start, then we are starting at point D and are going up. If the variable closest is equal to two and if ra_end is greater than ra_start, then we starting at point F and are going right. If the variable closest is equal to two and if ra_end is less than ra_start, then we are starting at point F and are going left. If the variable closest is equal to three and if dec_end is greater than dec_start, then we are starting at point H and we are going up. If the variable closest is equal to three and if dec_end is less than dec_start, then we are starting at point H and we are going down.

This is all done using an if then else if structure. In each case the variables scan_axis_size, cross_scan_axis_size and position_angle are specified in terms of the variables, x, y and scan_angle. Next the variable horizontal_range is set equal to the variable scan_axis_size. Then the variable vertical_range is set equal to the variable cross_scan_axis_szize.

Then pixel size in arcseconds is specified by the value of the variable pixel_size_in_arcseconds. pixel_size_in_arcseconds=2.0. Next we calculate the number of pixels we need in the image. The variable number_in_horizizontal_direction$\frac{pixel_size_in_arcseconds}{\text{is defined below,}}\cdot$

$$number\_in\_horizontal\_direction = \frac{horizontal\_range}{\frac{pixel\_size\_in\_arcseconds}{3600.0}}. \tag{A.9}$$

The variable number_in_vertical_direction is defined below,

$$number\_in\_vertical\_direction = \frac{vertical\_range}{\frac{pixel\_size\_in\_arcseconds}{3600.0}}. \tag{A.10}$$

These values are then converted to the nearest integer plus a safety factor just in case. The in house function nint does the job converting the input to the nearest integer as opposed to the largest integer less than the input value. The variable number_in_hozizontal_direction is redefined as,

$$number\_in\_horizontal\_direction = nint(number\_in\_horizontal\_direction) + 500. \tag{A.11}$$

The variable number_in_vertical_direction is redefined as,

$$number\_in\_vertical\_direction = nint(number\_in\_vertical\_direction) + 500. \tag{A.12}$$

Then the blank array pacs_raster_master_image is generated with two dimensions. The first dimension being equal to the variable number_in_horizontal_direction. The second dimension being equal to the variable number_in_vertical_direction. Next the in house subprogramme icmkhdr is used to make the header needed by the subprogramme coadd_timelines_into_master_pacs_image_version2.pro. The inputs are, number_in_hozizontal_direction, number_in_vertical_direction,

$-1*pixel\_size\_in\_arcseconds/3600.0d0$, $pixel\_size\_in\_arcseconds/3600.0d0$, mean(image17), mean(image18), and position_angle. The only output is header_pacs_raster_master_image, which is the required header.

For the red70 and green 70 data sets the variable closest is equal to two. So in this case we are starting at F and then going along FH initially. In this case we are going left. For the red71 and green71 data sets the variable closest is equal to three. So in this case we are starting at H and then going along HB initially. In this case we are going right. So with the data we have at present we have only tested two out the eight cases.

Inside the subprogramme coadd_timelines_into_master_pacs_image_version2.pro, a subprogramme coadd_data1_with_data2.pro, is defined first ahead of the main subprogramme. The inputs to the subprogramme coadd_data1_with_data2.pro, are, data1, noise1, data2 and noise2. The outputs from the subprogramme coadd_data1_with_data2.pro, are, result and noise_result.

If the variable data1 is finite, if the variable noise1 is finite, if the variable data2 is finite and if the variable noise2 is finite, then the variables result and noise_result are defined below.

$$result = \frac{\frac{data1}{noise1^2} + \frac{data2}{noise2^2}}{\frac{1}{noise1^2} + \frac{1}{noise2^2}} \tag{A.13}$$

$$noise\_result = \frac{1}{sqrt(\frac{1}{noise1^2} + \frac{1}{noise2^2})}. \tag{A.14}$$

If any of the inputs to the subprogramme coadd_data1_with_data2.pro, is not finite then the output variables result and noise_result are both zero.

Inside the subprogramme coadd_timelines_into_master_pacs_image_version2.pro, the inputs are image1, noise, image17, image18, master_image, master_noise, master_header and the optional inputs, xpix, ypix and npix. The subprogramme coadd_timelines_into_master_pacs_image_version2.pro, modifies the inputs master_image and master_noise.

In the third version of the PACS pipeline programme none of the optional inputs to the subprogramme coadd_timelines_into_master_pacs_image_version2.pro are specified. The input to the subprogramme image1 is a three dimensional array containing the timeline filtered PACS flux. The input to the subprogramme noise is a three dimensional array containing the timeline filtered PACS noise. The input to the subprogramme image17 is a three dimensional array giving the right ascension of each PACS pixel as a function of time. The input to the subprogramme image18 is a three dimensional array giving the declination of each PACS pixel as a function of time. The input to the subprogramme master_image is a blank two dimensional array modified by the subprogramme. The array master_image will contain the PACS flux projected onto the sky. The input to the subprogramme master_noise is a blank two dimensional array modified by the subprogramme. The array master_noise will contain the PACS noise projected onto the sky. The input to the subprogramme master_header is the header needed by the subprogramme to do the sky projecting.

The arrays xpix and ypix are the x and y pixel positions of the input right ascension and declination arrays. The variable npix is the number of detector pixels that are contributing to each pixel on the sky. The subprogramme uses the in house subprogramme adxy to convert pixel positions and does noise-weighted coadds. If the optional inputs are selected they return the appropriate arrays to the main programme. If the optional inputs are selected and both xpix and ypix are externally specified then the subprogramme uses these arrays and does not calculate them it self.

I will from now on only consider the case where the optional inputs to the subprogramme are not present as they are in the third version of the PACS pipeline programme. The in house subprogramme adxy has the inputs, master_header, image17 and image18. The outputs from adxy are the arrays xpix and ypix. The arrays xpix and ypix are both modified using the in house function nint, so that each their elements is changed to the nearest integer to that element.

If an element of the array master_noise is equal to zero then change that element to the value 1×10^{10}. The array s is the size of the array image17. The element s[0] is the number of dimensions of image17, in this case 3. The element s[1] is the size of the first dimension. The element s[2] is the size of the second dimension. The element s[3] is the size of the third dimension. The size of the array master_image is obtained and used to specify the number of detector pixels contributing to each pixel on the sky. This is a two dimensional array giving this number for each pixel on the sky. This is given by the two dimensional array number_of_pixels_contributing. This array has the same size and shape as the array master_image. At this stage it is a blank two dimensional array of integers.

Next we loop over the detector pixels to do the noise-weighted coadds. The positive or zero integer in the range zero to s[1]-1 ireadout is the variable that specifies the first dimension of the arrays image1 and noise. The positive or zero integer in the range zero to s[2]-1 ipixel is the variable that specifies the second dimension of the arrays image1 and noise. The positive or zero integer in the range zero to s[3]-1 jpixel is the variable that specifies the third dimension of the array image1 and noise.

We use three nested for loops to do the noise-weighted coadding. The outer most for loop uses the variable ipixel from zero to s[2]-1, as the loop variable. The middle for loop uses the variable jpixel from zero to s[3]-1, as the loop variable. The first value of ireadout which is zero is done first then for looping begins. Then the central for loop uses the variable ireadout from one to s[1]-1, as the loop variable.

Just inside the outermost for loop is the first of two commands calling the in house subprogramme printdone is present. The second and final command calling the in house subprogramme printdone is present at the programme line just after the end of the nested for loops. This in house subprogramme with inputs specified in this subprogramme prints to the screen the percent completed when each of the stages of the outer most for loops is completed. The output is given to the nearest 0.1%.

The array xpix is the value of the first dimension of master_image or master_noise, for each PACS pixel as a function of time. Hence xpix is a three dimensional array. The array ypix is the value of the second dimension of master_image or master_noise, for each PACS pixel as a function of time. Hence ypix is a three dimensional array.

Next the middle for loop begins just after the first of the two commands calling the in house subprogramme printdone. Then we coadd the first readout, ireadout=0, of this detector pixel into the master_image plus the master_noise. The variable signal_in_this_pixel_so_far is defined below,

$$signal\_in\_this\_pixel\_so\_far = master\_image[xpix[0, ipixel, jpixel], ypix[0, ipixel, jpixel]]. \tag{A.15}$$

The variable noise_in_this_pixel_so_far is defined below,

$$noise\_in\_this\_pixel\_so\_far = master\_noise[xpix[0, ipixel, jpixel], ypix[0, ipixel, jpixel]]. \tag{A.16}$$

The variable signal_to_average_in is defined below,

$$signal\_to\_average\_in = image1[0, ipixel, jpixel]. \tag{A.17}$$

The variable noise_to_average_in is defined below,

$$noise\_to\_average\_in = noise[0, ipixel, jpixel]. \tag{A.18}$$

Next the in house subprogramme given at the beginning of this subprogramme, coadd_data1_with_data2.pro is used. The inputs are signal_in_this_pixel_so_far, noise_in_this_pixel_so_far, signal_to_average_in and noise_to_average_in. The outputs are the variables combined_signal and combined_noise.

Then the array sections of the arrays, master_image and master_noise are redefined below,

$$master\_image[xpix[0, ipixel, jpixel], ypix[0, ipixel, jpixel]] = combined\_signal, \tag{A.19}$$

$$master\_noise[xpix[0, ipixel, jpixel], ypix[0, ipixel, jpixel]] = combined\_noise. \tag{A.20}$$

For all subsequent readouts, if the y position is constant, then interpolate the x positions. The interpolation is being done because the PACS instrument is scanning the field so fast that it is jumping pixels. This is okay because the data at this stage has already been smoothed along the timeline direction which is the x-axis in the master image. Now the central for loop runs over the ireadout values from one to s[1]-1, in steps of one. Just inside the central for loop we have the if structure where the interpolation is done, plus the subprogramme coadd_data1_with_data2.pro is used to generate the arrays master_image and master_noise for all the other ireadouts in the central for loop. Some of the interpolation requires another for loop inside the central for loop and inside the if structure. The subprogramme coadd_data1_with_data2.pro is one defined at the beginning of this subprogramme. If the y-position has jumped or if the x-position is stationary, then there is no interpolation to be done and it is just a standard coadd.

So as a result of the subprogramme coadd_timelines_into_master_pacs_image_version2.pro, the arrays master_image and master_noise have respectively the correct PACS flux and PACS noise for all of their elements. This means given a right ascension and declination of a position in the sky, we can use adxy, the master_header, the right ascension of the sky position and the declination of the sky position to get the pixel location for this sky position. This pair of inputs can be used to get the PACS flux and PACS noise for that sky position. This first input is the value of the first dimension needed for the master_image or master_noise arrays. The second input is the value of the second dimension needed for the master_image or master_noise arrays.

Inside the subprogramme make_kernel_for_cross_scan_filtering.pro, the inputs are the variables size_of_kernel and fwhm. Inside the subprogramme

make_kernel_for_cross_scan_filtering.pro the output is the array kernel. The variable size_of_kernel is the length of the kernel needed in the cross-scan direction to do the cross-scan filtering. The variable fwhm is the full width at half maximum needed to do the cross-scan filtering. The array kernel is a two dimensional array. The first dimension is equal to the number of rows which is in this case three. The second dimension is equal to the variable size_of_kernel. The variable number_of_rows is equal to three. The variable theta is defined below,

$$theta = \frac{fwhm}{2.0d0 * sqrt(alog(4.0d0))}. \tag{A.21}$$

The variable a is defined below,

$$a = \frac{1.0d0}{2.0d0 * theta^2}. \tag{A.22}$$

The variable middle is set equal to half that of the variable size_of_kernel. The array x is the array of real numbers which correspond to the integers in the range zero to size_of_kernel minus one. The array y is defined below,

$$y = exp(-a * (x - x[middle])^2). \tag{A.23}$$

The first row of the array *kernel* is set to zero every where. The second row of the array *kernel* is set equal to the one dimensional array y. Each element of the second row of the array *kernel* is set to the matching element of the array y. The third row of the array *kernel* is set to zero every where.

The kernel was defined in this manner because it is not clear what IDL would do with a one dimensional kernel used to filter two dimensional image.

I will now give the programme listings of the four programmes that make up the cross-scan filtering section of the latest version of the PACS pipeline programme. They are in order of use: make_sky_master_image_for_pacs_raster_version2.pro,

coadd_timelines_into_master_pacs_image_version2.pro, make_kernel_for_cross_scan_filtering.pro,

and scuba_src_ext.pro. The last programme was written by my supervisor.

```
pro make_sky_master_image_for_pacs_raster_version2, $
   image17, image18, $
   pacs_raster_master_image, header_pacs_raster_master_image, $
   debug=debug
;+
; NAME:
;      MAKE_SKY_MASTER_IMAGE_FOR_PACS_RASTER
;
;
; PURPOSE:
;      Make sky image for coadding PACS timelines into with correct pos angle
;
;
; CATEGORY:
;      ATLAS data reduction
;
;
;
; CALLING SEQUENCE:
;      make_sky_master_image_for_pacs_raster, ra_array, dec_array, $
;           output_image, output_header [,/debug]
;
;
; INPUTS:
;      image17: fltarr containing ra values of all pixels
;      image18: fltarr containing dec values of all pixels
;
;
; OPTIONAL INPUTS:
;      None
;
;
; KEYWORD PARAMETERS:
;      Debug: make some debugging checks that scan angle is OK
;
;
; OUTPUTS:
;      pacs_raster_master_image: fltarr master image template
;      header_pacs_raster_master_image: FITS header for master image
;
```

```
;
; OPTIONAL OUTPUTS:
;      None
;
;
; COMMON BLOCKS:
;      None
;
;
; SIDE EFFECTS:
;      None
;
;
; RESTRICTIONS:
;      Beta-test
;
;
; PROCEDURE:
;      Calculates scan angle then makes FITS image with correct
;      orientation on the sky for the scan angle.
;
;
; EXAMPLE:
;      make_sky_master_image_for_pacs_raster, $
;          image17, image18, $
;          pacs_raster_master_image, header_pacs_raster_master_image, $
;          /debug
;
; MODIFICATION HISTORY:
;      Written by Stephen Serjeant and Dominic Goodfellow,
;      18th February 2011. Any modifications to be described here.
;
;
;-


;;;
;;; Find the scan angle from first 1000 readouts
;;;

ra_start = image17[0,0,0]
ra_end = image17[999,0,0]

dec_start = image18[0,0,0]
dec_end = image18[999,0,0]
```

```
delta_dec = dec_end - dec_start
delta_ra = (ra_end - ra_start)*cos(!dtor*0.5*(dec_start+dec_end))

;scan_angle = atan(delta_dec,delta_ra)/!dtor
scan_angle = atan(abs(delta_dec),abs(delta_ra))/!dtor


;;;
;;; Then find the required size of the image
;;;

ramin = min(image17)
ramax = max(image17)
decmin = min(image18)
decmax = max(image18)
typical_dec = 0.5*(decmin+decmax)

wdecmin=where(image18 eq decmin,nwdecmin)
wdecmax=where(image18 eq decmax,nwdecmax)
wramin=where(image17 eq ramin,nwramin)
wramax=where(image17 eq ramax,nwramax)
rawdecmin=image17[wdecmin[0]]
rawdecmax=image17[wdecmax[0]]
decwramin=image18[wramin[0]]
decwramax=image18[wramax[0]]

;;; a,b,c,d are lengths in decimal degrees
a = (rawdecmin - ramin)*cos(typical_dec*!dtor)
b = (ramax - rawdecmin)*cos(typical_dec*!dtor)
c = decwramax - decmin
d = decmax - decwramax

x = sqrt(d^2 + a^2)
y = sqrt(b^2 + c^2)

;;; NB we have already defined ra_start, dec_start, ra_end, dec_end
;;; for the first scan of this raster. NB also: ra_end etc is not
;;; the end of the scan but rather some intermediary point.

;;; Make a rough estimate of the positions of the raster corners,
;;; labelled B, D, F, H in Dominic's lab book.

ra_B = ramin + a/cos(typical_dec*!dtor)
dec_B = decmin
```

```
ra_D = ramax
dec_D = decmin + c

ra_F = ramin + b/cos(typical_dec*!dtor)
dec_F = decmax

ra_H = ramin
dec_H = decmin + d

gcirc,1,ra_start/15.0d0,dec_start,[ra_B,ra_D,ra_F,ra_H]/15.0d0, $
[dec_B,dec_D,dec_F,dec_H],distance
closest = (where(distance eq min(distance)))[0]
;; if closest=0, then we start at point B.
;; if closest=1, then we start at point D.
;; if closest=2, then we start at point F.
;; if closest=3, then we start at point H.

;; NB THIS ALGORITHM ASSUMES WE ARE NOT STRADDLING RA=0 HOURS.
if(closest eq 0) then begin
;; we start at point B.
   if(ra_end gt ra_start) then begin
;;    we are starting at point B and going right
      scan_axis_size = y
      cross_scan_axis_size = x
      position_angle = scan_angle
   endif else begin
;;    we are starting at point B and going left
      scan_axis_size = x
      cross_scan_axis_size = y
      position_angle = 180.0 - scan_angle
   endelse
endif else begin
   if(closest eq 1) then begin
;;    we start at point D.
      if(dec_end lt dec_start) then begin
;;       we are starting at point D and going down.
         scan_axis_size = y
         cross_scan_axis_size = x
         position_angle = scan_angle
      endif else begin
;;       we are starting at point D and going up.
         scan_axis_size = x
         cross_scan_axis_size = y
         position_angle = 180.0 - scan_angle
      endelse
   endif else begin
      if(closest eq 2) then begin
;;       we start at point F.
         if(ra_end gt ra_start) then begin
;;          we are starting at point F and going right.
            scan_axis_size = x
            cross_scan_axis_size = y
            position_angle = 180.0 - scan_angle
         endif else begin
;;          we are starting at point F and going left.
            scan_axis_size = y
            cross_scan_axis_size = x
            position_angle = scan_angle
         endelse
      endif else begin
;;       we start at point H.
         if(dec_end gt dec_start) then begin
;;          we are starting at point H and going up.
            scan_axis_size = y
            cross_scan_axis_size = x
            position_angle = scan_angle
         endif else begin
;;          we are starting at point H and going down.
            scan_axis_size = x
            cross_scan_axis_size = y
            position_angle = 180.0 - scan_angle
         endelse
      endelse
   endelse
endelse



;if(scan_angle gt 0 and scan_angle le 90.0) then begin
;   scan_axis_size = y
;   cross_scan_axis_size = x
;   position_angle = -scan_angle
;   if(keyword_set(debug)) then print,'case 1'
;endif else begin
;   if(scan_angle gt 90.0) then begin
;      scan_axis_size = x
;      cross_scan_axis_size = y
;      position_angle = 180.0 - scan_angle
;      if(keyword_set(debug)) then print,'case 2 [UNTESTED]'
;   endif else begin
;      if(scan_angle le 0 and scan_angle gt -90.0) then begin
;         scan_axis_size = x
;         cross_scan_axis_size = y
;         position_angle = -scan_angle
;         if(keyword_set(debug)) then print,'case 3'
;      endif else begin
;         scan_axis_size = y
;         cross_scan_axis_size = x
;         position_angle = 90.0 + scan_angle
;         if(keyword_set(debug)) then print,'case 4 [UNTESTED]'
;      endelse
;   endelse
;endelse

horizontal_range = scan_axis_size
vertical_range = cross_scan_axis_size

;;;
;;; Then decide the pixel size we want in arcseconds
;;;

pixel_size_in_arcseconds = 2.0

;;;
;;; Then calculate the number of pixels we need in the image
;;;

number_in_horizontal_direction = horizontal_range / $
                                 (pixel_size_in_arcseconds/3600.0)
number_in_vertical_direction = vertical_range / $
                                 (pixel_size_in_arcseconds/3600.0)

;;;
;;; Make an integer and add a bit just to be on the safe side
;;;

;number_in_horizontal_direction = nint(number_in_horizontal_direction)+200
;number_in_vertical_direction = nint(number_in_vertical_direction)+200
number_in_horizontal_direction = nint(number_in_horizontal_direction)+500
number_in_vertical_direction = nint(number_in_vertical_direction)+500

;;;
;;; Define an image and a header
;;;
```

```
pacs_raster_master_image = $
   dblarr(number_in_horizontal_direction,number_in_vertical_direction)

;icmkhdr, number_in_horizontal_direction, number_in_vertical_direction, $
;         -1*pixel_size_in_arcseconds/3600.0d0, $
;         pixel_size_in_arcseconds/3600.0d0,$
;         mean(image17), mean(image18), scan_angle, $
;         header_pacs_raster_master_image


icmkhdr, number_in_horizontal_direction, number_in_vertical_direction, $
         -1*pixel_size_in_arcseconds/3600.0d0, $
         pixel_size_in_arcseconds/3600.0d0,$
         mean(image17), mean(image18), position_angle, $
         header_pacs_raster_master_image

;;;
;;; Test that this image and header is correctly oriented
;;;

if(keyword_set(debug)) then begin
   icplot, pacs_raster_master_image, header_pacs_raster_master_image
   icellipse_oplot, header_pacs_raster_master_image, $
                    image17[0:999,0,0], image18[0:999,0,0],/psym
   icellipse_oplot, header_pacs_raster_master_image, $
                    image17[10000L:10999L,0,0], $
                    image18[10000L:10999L,0,0],/psym
endif

;;;
;;; Closing
;;;
;stop
end


pro coadd_data1_with_data2, data1, noise1, data2, noise2, result, noise_result

if(finite(data1) eq 1 and finite(noise1) eq 1 and $
   finite(data2) eq 1 and finite(noise2) eq 1) then begin
    result = $
      (data1/noise1^2 + data2/noise2^2) / $
      (1/noise1^2 + 1/noise2^2)
    noise_result = $
```

```
      1 / sqrt(1/noise1^2 + 1/noise2^2)
endif

end
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
pro coadd_timelines_into_master_pacs_image_version2, image1, noise, $
image17, image18, master_image, master_noise, master_header,xpix=xpix,$
ypix=ypix,npix=number_of_pixels_contributing

;+
; NAME:
;   COADD_TIMELINES_INTO_MASTER_PACS_IMAGE
;
;
; PURPOSE:
;   Coadds the PACS timelines into the master sky image
;
;
; CATEGORY:
;   ATLAS data reduction
;
;
; CALLING SEQUENCE:
;   coadd_timelines_into_master_pacs_image, image1, noise, ra, dec, $
;        master_image, master_noise, master_header
;
;
; INPUTS:
;   image1: PACS timeline data
;   noise: noise on PACS timeline data
;   image17: RA array of PACS timeline data
;   image18: Dec array of PACS timeline data
;   master_image: sky image to coadd into
;   master_noise: noise for sky image to coadd into
;   master_header: header for sky image to coadd into
;
;
; OPTIONAL INPUTS:
;   None
;
;
; KEYWORD PARAMETERS:
;   None
;
;
```

```
; OUTPUTS:
;   None (but modifies input master_image and master_noise)
;
;
; OPTIONAL OUTPUTS:
;   xpix,ypix: x and y pixel positions of the input ra and dec arrays
;   npix: number of detector pixels contributing to each pixel on the sky
;
;
; COMMON BLOCKS:
;   None
;
;
; SIDE EFFECTS:
;   Modifies input master_image and master_noise
;
;
; RESTRICTIONS:
;   Untested!
;
;
; PROCEDURE:
;   Uses adxy to convert to pixel positions and does noise-weighted coadds.
;
;
; EXAMPLE:
;  coadd_timelines_into_master_pacs_image, image1, noise, image17, image18, $
;      master_image, master_noise, master_header
;
; MODIFICATION HISTORY:
;  Written by Stephen Serjeant and Dominic Goodfellow, 18 Feb 2011.
;  STILL UNTESTED AS YET!
;
;
;-

;;; find pixel positions of pacs data
if(n_elements(xpix) eq 0 or n_elements(ypix) eq 0) then begin
        print,'Calculating xpixel ypixel positions...'
adxy, master_header, image17, image18, xpix, ypix
xpix = nint(xpix)
ypix = nint(ypix)
        print,'...done. '
        print,'Minmax(xpix)= ',minmax(xpix)
        print,'Minmax(ypix)= ',minmax(ypix)
        print,'Size(master_image)= ',size(master_image)
endif

;;; Put huge values of noise where there's no data
bad = where(master_noise eq 0, nbad)
if(nbad gt 0) then master_noise[bad] = 1d10

s = size(image17)

size_of_master_image = size(master_image)
number_of_pixels_contributing = intarr(size_of_master_image[1],size_of_master_image[2])

;ipixel = 31L
;jpixel = 6L
;ipixtest = [0,31]
;jpixtest = [4,6]
;
;for itest=0,n_elements(ipixtest)-1 do begin
;    ipixel = ipixtest[itest]
;    jpixel = jpixtest[itest]

;;; loop over the detector pixels
for ipixel = 0L, s[2]-1L do begin
    printdone,ipixel,1.0*s[2]
;    print,'Minmax(xpix)= ',minmax(xpix)
    for jpixel = 0L, s[3]-1L do begin

;;;         Coadd the first readout of this detector pixel into the
;;;         master image
        signal_in_this_pixel_so_far = master_image[xpix[0,ipixel,$
jpixel],ypix[0,ipixel,jpixel]]
        noise_in_this_pixel_so_far = master_noise[xpix[0,ipixel,jpixel],$
ypix[0,ipixel,jpixel]]
        signal_to_average_in = image1[0,ipixel,jpixel]
        noise_to_average_in = noise[0,ipixel,jpixel]
        coadd_data1_with_data2, $
          signal_in_this_pixel_so_far, noise_in_this_pixel_so_far,$
          signal_to_average_in, noise_to_average_in, $
          combined_signal, combined_noise
        master_image[xpix[0,ipixel,jpixel],ypix[0,ipixel,jpixel]] = $
combined_signal
        master_noise[xpix[0,ipixel,jpixel],ypix[0,ipixel,jpixel]] = $
combined_noise

;;;      For all subsequent readouts, if the y position is
;;;      constant, then interpolate the x positions. This is OK
;;;      because the data at this stage has already been smoothed
;;;      along the timeline direction which is the x-axis in the
;;;      master image.
        for ireadout = 1L,s[1]-1L do begin
;            stop
            if(xpix[ireadout,ipixel,jpixel] ne xpix[ireadout-1L,ipixel,jpixel]$
 and  ypix[ireadout,ipixel,jpixel] eq ypix[ireadout-1L,ipixel,jpixel])$
 then begin

;                n = abs(xpix[ireadout,ipixel,jpixel] - xpix[ireadout-1L,ipixel,jpixel])
;                xmin = min([xpix[ireadout,ipixel,jpixel],xpix[ireadout-1L,$
ipixel,jpixel]])
;                xinterpolation = xmin + lindgen(n)
                n = abs(xpix[ireadout,ipixel,jpixel] - xpix[ireadout-1L,ipixel $
,jpixel])

                if(xpix[ireadout,ipixel,jpixel] gt xpix[ireadout-1L,ipixel,$
jpixel]) then begin
                   xinterpolation = xpix[ireadout-1L,ipixel,jpixel] + 1 +$
 lindgen(n)
                endif else begin
                   xinterpolation = xpix[ireadout-1L,ipixel,jpixel] - 1 -$
 lindgen(n)
                endelse


                interpolated_signal = interpol(image1[[ireadout-1L,ireadout],$
ipixel,jpixel], xpix[[ireadout-1L,ireadout],ipixel,jpixel],xinterpolation)
                interpolated_noise = interpol(noise[[ireadout-1L,ireadout],$
ipixel,jpixel],xpix[[ireadout-1L,ireadout],ipixel,jpixel],xinterpolation)
;                if(ireadout eq 210173L or ireadout eq 214300L) then stop
                for iinterp=0L,n_elements(xinterpolation)-1L do begin
                    signal_in_this_pixel_so_far = master_image[xinterpolation $
[iinterp],ypix[ireadout,ipixel,jpixel]]
                    noise_in_this_pixel_so_far = master_noise[xinterpolation $
[iinterp],ypix[ireadout,ipixel,jpixel]]
                    signal_to_average_in = interpolated_signal[iinterp]
                    noise_to_average_in = interpolated_noise[iinterp]
                    coadd_data1_with_data2, $
                      signal_in_this_pixel_so_far, noise_in_this_pixel_so_far,$
                      signal_to_average_in, noise_to_average_in, $
                      combined_signal, combined_noise
                    master_image[xinterpolation[iinterp],ypix[ireadout,ipixel,$
jpixel]] = combined_signal
```

```
                    master_noise[xinterpolation[iinterp],ypix[ireadout,ipixel,$
jpixel]] = combined_noise
                    number_of_pixels_contributing[xinterpolation[iinterp],$
ypix[ireadout,ipixel,jpixel]] = number_of_pixels_contributing $
[xinterpolation[iinterp],ypix[ireadout,ipixel,jpixel]] + 1
;                    stop
                endfor
            endif else begin
 ;;;            If the y-position has jumped or if the x position
 ;;;            is stationary, then there's no interpolation to be
 ;;;            done and it's just a standard coadd.
                signal_in_this_pixel_so_far = $
                  master_image[xpix[ireadout,ipixel,jpixel],ypix[ireadout,$
ipixel,jpixel]]
                noise_in_this_pixel_so_far = $
                  master_noise[xpix[ireadout,ipixel,jpixel],ypix[ireadout,$
ipixel,jpixel]]
                signal_to_average_in = image1[ireadout,ipixel,jpixel]
                noise_to_average_in = noise[ireadout,ipixel,jpixel]
                coadd_data1_with_data2, $
                  signal_in_this_pixel_so_far, noise_in_this_pixel_so_far,$
                  signal_to_average_in, noise_to_average_in, $
                  combined_signal, combined_noise
                master_image[xpix[ireadout,ipixel,jpixel],ypix[ireadout,$
ipixel,jpixel]] = combined_signal
                master_noise[xpix[ireadout,ipixel,jpixel],ypix[ireadout,$
ipixel,jpixel]] = combined_noise
;                stop
            endelse
        endfor
    endfor
endfor
printdone,ipixel,1.0*s[2],/newline

;stop
end

pro make_kernel_for_cross_scan_filtering,size_of_kernel,fwhm,kernel
;fwhm, the full width at half maximum in pixels of the kernel.
;theta, is the standard deviation of the kernel with fwhm, as the full width
; at half maximum.
number_of_rows=3
theta=fwhm/(2.0d0*sqrt(alog(4.0d0)))
a=1.0d0/(2.0d0*theta^2)
;print,'Number of rows:',number_of_rows
```

```
;print,'theta=',theta
;print,'a=',a
middle=size_of_kernel/2
x=findgen(size_of_kernel)
y=exp(-a*(x-x[middle])^2)
kernel=dblarr(number_of_rows,size_of_kernel)
kernel[0,*]=0.0d0
kernel[1,*]=y
kernel[2,*]=0.0d0
end

pro scuba_src_ext, input_image, input_noise, psf, $
                   bestfitflux, delta_bestfitflux, chi2, dof, prob, $
                   no_ft=no_ft, silent=silent, $
                   c_iw_p=c_iw_p, $
                   c_w_p2=c_w_p2

;+
; NAME:
;       SCUBA_SRC_EXT
;
; PURPOSE:
;       Optimal noise-weighted (min-chi^2) point source filter with chi2 calc
;
; CATEGORY:
;       Image processing
;
; CALLING SEQUENCE:
; scuba_src_ext, image, noise, psf, bestfitflux, delta_bestfitflux, chi2
;
; INPUTS:
; image: (fltarr) image to have sources extracted
; noise: (fltarr) noise level in image
; Regions where noise=0 are assumed to have no data.
; psf: (fltarr) point spread function. There are no restrictions
; on the psf - eg can have negative values, and need not be
; normalised. The output flux is in units of multiples of
; the psf.
;
; OPTIONAL INPUTS:
;       c_iw_p, c_w_p2: internal convolutions - can input if already
;                       calculated previously to speed up routine
;
; KEYWORD PARAMETERS:
; no_ft: dont do convolutions with fourier transforms
; silent: dont keep printing percentage of calculation done
;
; OUTPUTS:
; bestfitflux: min-chi^2 flux of sources (expressed as a multiple
; of the PSF)
; delta_bestfitflux: error in flux
;
; OPTIONAL OUTPUTS:
; chi2: chi2 of fit
; dof: degrees of freedom of fit
; prob: likelihood of fit (high probs => acceptable model,
;  low probs => unacceptable model).
;
; COMMON BLOCKS:
; none
;
; SIDE EFFECTS:
; none
;
; RESTRICTIONS:
;       Copyright Stephen Serjeant.
;
; PROCEDURE:
; Performs min-chi^2 fit to image and noise, given the PSF.
; The resulting signal:noise map (bestfitflux/delta_bestfitflux)
; can be thresholded for a formally optimal point source
; extraction.
; Uses convolutions for speed.
;
; EXAMPLE:
; scuba_src_ext, image, noise, psf, flux, deltaflux
;
; MODIFICATION HISTORY:
;       Written by:     Stephen Serjeant (ICSTM) December 1998
;       Any modifications to be described here.
;
;       13 Jan 1999: Added checks for non-finite values in image
;               Steve Serjeant
; 14 Jan 1999: Added optional chi2, dof and prob outputs S Serjeant
; 22 Apr 1999: Non-finite values in outputs now set to zero S Serjeant
;       14 Aug 2000: Added c_iw_p, c_w_p2 optional inputs S Serjeant v2.0
;       13 Mar 2001: Converted to idl 5.4 S.Serjeant v2.1
;
;-
```

```
VERSION='2.1'

on_error,2

;-----------------------------------------------
;
; Parameter checks
;
;-----------------------------------------------

if(n_params() lt 5 or n_params() gt 8) then begin
print,'USAGE: scuba_src_ext, image, noise, psf, $'
print,'   flux, deltaflux [, chi2, dof, prob, /no_ft, /silent,$'
        print,'   c_iw_p=c_iw_p, c_w_p2=c_w_p2]'
return
endif

if(n_params() gt 5) then do_chi2=1B else do_chi2=0B
if(n_params() eq 8) then do_prob=1B else do_prob=0B


;-----------------------------------------------
;
; Embed inputs into larger images.
; Necessary because the convolve function edge-wraps, but
; the alternative convol function is slower.
;
;-----------------------------------------------

sim=size(input_image)
spsf=size(psf)

image = fltarr(sim[1]+4*spsf[1],sim[2]+4*spsf[2])
noise = fltarr(sim[1]+4*spsf[1],sim[2]+4*spsf[2])

image[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]=input_image[*]
noise[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]=input_noise[*]

if(do_chi2) then begin
coverage = 0.0*image
coverage[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]=1
endif

;-----------------------------------------------
;
```

```
; Check for regions which need to be masked
;
;-------------------------------------------------

;;; First check for non-finite values
bad = where((finite(image) eq 0) or (finite(noise) eq 0), nbad)
if(nbad gt 0) then begin
image[bad]=0
noise[bad]=0
if(do_chi2) then coverage(bad)=0
message,'WARNING: non-finite values in input images (will be masked)',/info
endif

;;; Mask all non-finite values, and any other regions where noise=0
badpix = where(noise eq 0, nbad)
if(nbad gt 0) then noise[badpix] = 1e10*max(noise)
if(nbad gt 0) then image[badpix] = 0.0
if(nbad gt 0 and do_chi2) then coverage[badpix] = 0.0

;----------------------------------------------
;
; Perform the convolutions equivalent to the
; min-chi2 fitting
;
;----------------------------------------------

snr = image / noise
weight = 1/noise^2

if(n_elements(c_iw_p) eq 0) then $
  c_iw_p = convolve(image*weight,psf,no_ft=no_ft)         ;;; Define useful
; convolutions

if(n_elements(c_w_p2) eq 0) then $
  c_w_p2 = convolve(weight,psf^2,no_ft=no_ft)
bestfitflux = c_iw_p / c_w_p2                   ;;; Best fit flux

okfit = where(c_w_p2 gt 0)
delta_bestfitflux = 0.0*c_w_p2
delta_bestfitflux[okfit] = 1.0/sqrt(c_w_p2[okfit])        ;;; error on best
;fit flux



if(do_chi2) then begin
```

```
chi2_0 = convolve(snr^2, 0*psf+1,no_ft=no_ft) ;;; Chi2 of flux=0 model
chi2 = (chi2_0 - bestfitflux*c_iw_p)>0       ;;; Chi2 of best fit flux
; model
dof=convolve(coverage,0.0*psf+1,no_ft=no_ft) - 1 ;;; Degrees of
;freedom in chi2s
endif

;---------------------------------------------
;
; Restore original image segments
;
;---------------------------------------------

snr    = snr[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
noise  = noise[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
image  = image[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
bestfitflux $
       = bestfitflux[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
delta_bestfitflux $
       = delta_bestfitflux[$
 2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
if(do_chi2) then begin
chi2   = chi2[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
dof    = dof[2*spsf[1]:2*spsf[1]+sim[1]-1,2*spsf[2]:2*spsf[2]+sim[2]-1]
endif

if(do_prob) then prob = 1-chisqr_pdf_array(chi2,dof,silent=silent)

;---------------------------------------------
;
; Reset any non-finite values in outputs
;
;---------------------------------------------

bad = where(finite(bestfitflux) eq 0, nbad)
if(nbad gt 0) then bestfitflux[bad]=0
bad = where(finite(delta_bestfitflux) eq 0, nbad)
if(nbad gt 0) then delta_bestfitflux[bad]=0

if(do_chi2) then begin
bad = where(finite(chi2) eq 0, nbad)
if(nbad gt 0) then chi2[bad]=0
bad = where(finite(dof) eq 0, nbad)
if(nbad gt 0) then dof[bad]=0
endif
```

```
if(do_prob) then begin
bad = where(finite(prob) eq 0, nbad)
if(nbad gt 0) then prob[bad]=0
endif

return
end
```