

# **DETECTION OF ANOMALIES IN THE THERMAL INFRARED EMISSIONS PRIOR TO TEN EARTHQUAKES IN IRAN**

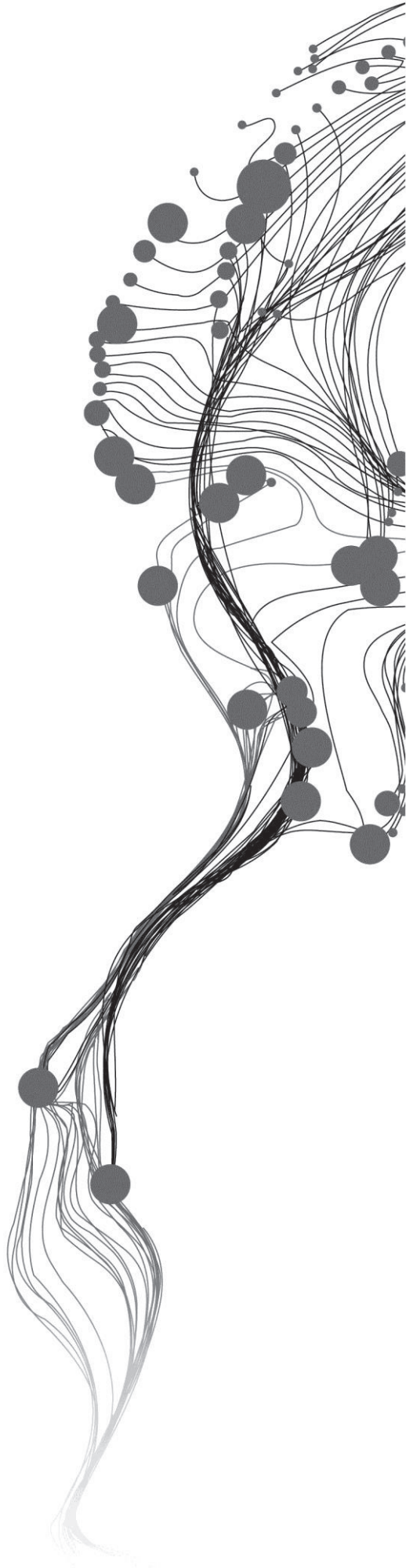
MARISOL G. LÓPEZ MEDA

[March, 2014]

SUPERVISORS:

Dr. Mark van der Meijde. Associate professor. Dpt. Of Earth System Analysis

Dr. Chris Hecker. Lecturer/Researcher. Dpt. Of Earth System Analysis.



# **DETECTION OF ANOMALIES IN THE THERMAL INFRARED EMISSIONS PRIOR TO TEN EARTHQUAKES IN IRAN**

MARISOL G. LÓPEZ MEDA

Enschede, The Netherlands, [March, 2014]

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Applied Earth Sciences

## **SUPERVISORS:**

Dr. Chris Hecker. Lecturer/Researcher. Dpt. Of Earth System Analysis.

Dr. Mark van der Meijde. Associate professor. Dpt. Of Earth System Analysis

## **THESIS ASSESSMENT BOARD:**

Prof. Dr. Victor G. Jetten (Chair)

Prof. Dr. Rens van Beek (Universiteit Utrecht)

#### DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

# ABSTRACT

Earthquakes are one of the most devastating natural hazardous events affecting in a global scale and with no warning. That makes the study of earthquakes a topic of worldwide interest. Due to its impact, the study of different phenomenon able to warn for the present of future seismic activity have been developed through the years. A phenomena considerer in literature for preceding seismic events, is the anomalous rise of ground temperature. However, are so many factors involving this phenomena that is still challenging the link of anomalous increase in the ground temperature and the presence of a future earthquake.

The present study was developed with the aim of verify the presence of anomalies in the Thermal Infrared emissions from areas covering ten seismic events previously reported for having the presence of anomalies prior their occurrence. The events where previously reported by Saraf et al., 2008 using Land Surface Temperature maps with NOAA-AVHRR observations. Saraf et al., 2008 documented to observe a dual anomalous peak in the temperature for seismic events with magnitud below 6 and a single peak for events with magnitude 6 or above. However, the findings leave questions regarding the extent of the time series used for observation and the spatial characteristics of the found anomalies. Saraf et al., 2008 used a short time series which was decreasing due to cloud coverage and meteorological effect in the images.

Aiming to overcome those limitations, the present study was carried out applying an existing methodology formerly developed by ITC team. Khan, 2010, Buzzo, 2012, Pavlidou, 2013 documented a more sophisticated and unbiased methodology which uses a cloud masking and a pixel normalization technique in the attempt ro remove the cloud coverture and neutralize the meteorological effect in the images. The existing methodology works with hypertemporal data which allows a time series analysis with more statistical relevance.

When comparing results from the present study with previously reported by Saraf et al., 2008, the large discrepancy show a more critical view in time and space of the presence of anomalies from the results of the present study. The use of a larger time series for observation show to discard as anomalous some of the findings made using a short time series.

## **Keywords:**

*Earthquake precursors, Thermal anomaly, Time series analysis*



## ACKNOWLEDGEMENTS

First on my list are my supervisors: Mark van der Meijde and Chris Hecker. I am deeply grateful for their constant patience, their sense of humour and the amount of things they taught me. For being there and always having the door open for me. And most of all, for giving me the opportunity to work on such challenging and fascinating topic.

I would like to thank the ESA department and whole ITC for all the facilities and knowledge provided during the learning phase of this MSc research.

I am deeply thankful for the unique experience, for meeting and working with wonderful people from all around the world.

Special thanks to Effie, who as a good friend and lover of the topic, was always willing to help and give her best suggestions even at the last minute in all matters. For her feedback and support.

Thanks to all my ITC family, for being a support in this one and a half year.

I would also like to thank my parents who are always supporting and encouraging me with their best wishes in any project I try. Without them, it just wouldn't be possible.

# TABLE OF CONTENTS

---

1.	Introduction.....	7
1.1.	Remotely sensed thermal infrared data.....	8
1.2.	Existing methodology.....	11
1.2.1.	Import.....	11
1.3.	Cloud masking.....	11
a	Relative-Absolute Mask (RAM).....	11
b	Polynomial Fit (PF) mask.....	11
1.3.2.	Anomaly detection.....	12
a	Pixel normalization.....	12
b	Anomaly flagging.....	12
1.4.	Research objectives.....	13
1.4.1.	Main Objective.....	13
1.4.2.	Specific objectives.....	13
1.5.	Research Questions.....	13
2.	Methodology.....	14
2.1.	Datasets.....	15
2.2.	Import.....	16
2.3.	Cloud masking.....	16
2.3.1.	Polynomial-Fit iterative method.....	16
2.3.2.	Anomaly detection.....	17
a	Pixel normalization.....	17
b	Anomaly flagging.....	17
3.	Results.....	18
3.1.	Methodology review and technical considerations.....	18
3.1.1.	Import.....	18
3.1.2.	Polynomial-Fit iterative method.....	18
3.1.3.	Pixel normalization.....	21
3.1.4.	Anomaly flagging.....	22
a	Dataset 1.....	22
	Bam earthquake (01:56 UTC, December 26 <sup>th</sup> , 2003).....	23
	Kerman earthquake (04:02 UTC, August 21 <sup>st</sup> , 2003).....	27
	Zarand earthquake (02:25 UTC, February 22 <sup>nd</sup> , 2005).....	31
b	Dataset 2.....	35
	Jahron (17:40 UTC, July 10 <sup>th</sup> , 2003).....	35
	Qeshm (10:22 UTC, November 27 <sup>th</sup> , 2005).....	37
	Faryab (07:31 UTC, February 28 <sup>th</sup> , 2006).....	37
	Fin (07:28 UTC, March 25 <sup>th</sup> , 2006).....	39
	Persian Gulf (21:02 UTC, June 28 <sup>th</sup> , 2006).....	41
c	Dataset 3.....	42
	Changureh-Avaj (02:58 UTC, June 22 <sup>nd</sup> , 2002).....	43
	Firozabad-Kajoor (12:38 UTC, May 28 <sup>th</sup> , 2004).....	46
4.	Discussion.....	48

## LIST OF FIGURES

---

Figure 2.1. Flowchart of the methodological steps followed in the present study.....	14
Figure 2.2. Delimitation and coverture of datasets used for the present study. Dataset 1, 2 and 3 are shown in red, green and yellow respectively. Google earth caption. ....	15
Figure 3.1a-d. Observation on 16:00-19-12-2003. Pixel (66,82) is indicated in yellow. Removed clouds are shown in red. (a) Image showing imported data, no further processing. (b) Output from applying the PF mask as proposed by Buzzo (2012). (c) Output from applying two iterations of the PF mask. (d) Output from applying three iterations of the PF mask. Green arrows point (c-d) to clouds with visible poor delimitation in the edges.....	19
Figure 3.2a-d. Time series for pixel (66,82). Vertical red line indicates the observation time (16:00-19-12-2003) for figures (3.1a-d). Lines in black shown the statistics of the imported data (a). Red row points a very low value possible due to an error from the satellite. For comparison of the results, an overlay of the PF mask and the Polynomial-fit iterative method is shown (a-d). Lines in blue correspond to the PF applied to the data as proposed by Buzzo (Buzzo). Lines in magenta correspond to the output from applying a second iteration of the PF mask. Lines in green correspond to the output from applying a third iteration of the PF mask.....	20
Figure 3.3. Statistics of pixel (66,82) after pixel normalization. Graph show daily morning patterns due to fog between 07:00 and 09:00 hrs. Red arrow pointing to a couple of the patterns. Blue arrow indicated a very low value due to the normalization of foggy/cloudy pixels. ....	21
Figure 3.4. 02:00-26-12-2003 observation from dataset 1 (see 2.1) after correcting for missing values. Pixels in yellow correspond to earthquake epicentres: (1) Kerman; (2) Bam; (3) Zarand; (4) Faryab; (5) Fin. Pixel size exaggerated by 3x3 for better visualization.....	22
Figure 3.5a-g. Images showing observations at 02:00hrs. In yellow the Bam and Kerman epicentre. Pixel (68,86) shown in red. Pixels in blue count 80-100% of the absolute maximum anomaly count for the whole time series. Pixels in magenta count 50-79%. Pixels in green count 30-49%. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%. ....	24
Figure 3.6a-b. Graphs showing anomaly count for pixel (68,86). (a) Anomaly count over the whole time series. (b) Anomaly count for year 2003. Peak with the maximum count (on 19 Dec 2003), is pointed with a black arrow. Red line indicates the time of occurrence for the Bam earthquake (Dec 26th, 2003. 01:56hrs (UTC)). ....	25
Figure 3.7a-b. Observations taken from Saraf et al. (2008) reported results for the Bam earthquake using LST maps. The epicentre of the earthquake is represented by a star in both images. Areas with higher temperature are coloured in red. ....	26
Figure 3.8a-e. Images showing observations at 04:00hrs. In yellow the Kerman and Bam epicentre oversized by 3x3 for better visualization. Pixel (118,69) shown in red. Pixels in magenta count 50-79% of the absolute maximum anomaly count for the whole time series. Pixels in green count 30-49%. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%. ....	28
Figure 3.9a-b. Graphs showing anomaly count for pixel (118,69). (a) Anomaly count over the whole time series. (b) Anomaly count for year 2003. Peak with the maximum count over the whole time series, is pointed with a black arrow. Red line indicates the time of occurrence for the Kerman earthquake (Aug 21st, 2003. 04:02hrs (UTC)). ....	29
Figure 3.10a-b. Observations taken from Saraf et al. (2008) reported results for the Kerman earthquake using LST maps. The epicentre of the earthquake is represented by a star in both images. Areas with higher temperature are coloured in red. ....	30

Figure 3.11a-g. Images showing observations at 02:30hrs. In yellow Zarand epicentre, pixel (46,46). Pixels in magenta count 50-79% of the absolute maximum anomaly count for the whole time series. Pixels in green count 30-49%. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%.....	32
Figure 3.12. Graph showing the anomaly count for the Zarand epicentre, pixel (46,46). Yearly occurring patterns are identifiable through the whole time series. Red line indicates the time for the Zarand earthquake (Feb 22 <sup>nd</sup> , 2005. 02:25 (UTC)).....	33
Figure 3.13a-d. Observations taken from Saraf et al. (2008) reported results for the Zarand earthquake using LST maps. The epicentre of the earthquake is represented by a star in both images. Areas with higher temperature are coloured in red.....	34
Figure 3.14. Image showing observation from dataset 2 when the Jahron earthquake occurred. Image already corrected for missing values for the whole time series. Pixels in yellow correspond to earthquake epicentres: (1) Jahron; (2) Qeshm; (3) Faryab; (4) Fin; (5) Persian Gulf. Pixel size exaggerated by 3x3 for better visualization.....	35
Figure 3.15a-g. Images showing observations at 17:30hrs. In yellow Jahron epicentre, pixel (35,43). In red pixel (17,35). Pixels in green count 30-49% of the absolute maximum anomaly count for the whole time series. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%.....	36
Figure 3.16a-g. Images showing observations at 07:30 hrs. In yellow the epicentre for the Fin and Faryab earthquakes due to their proximity in time and space. Pixels in blue count 80-100% of the absolute maximum anomaly count for the whole time series. Pixels in magenta count 50-79%. Pixels in green count 30-49%. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%. Dashed yellow line, delineates part of the coast for the Persian Gulf. ....	38
Figure 3.17a-g. Images showing observations at 07:30 hrs. In yellow the epicentre for the Fin and Faryab earthquakes due to their proximity in time and space. Pixels in blue count 80-100% of the absolute maximum anomaly count for the whole time series. Pixels in magenta count 50-79%. Pixels in green count 30-49%. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%. Dashed yellow line, delineates part of the coast for the Persian Gulf. ....	40
Figure 3.18. Image showing observation from dataset 3 when the Changureh-Avaj earthquake occurred. Image already corrected for missing values for the whole time series. Pixels in yellow correspond to earthquake epicentres: (1) Changureh-Avaj; (2) Firozabad-Kajoor. Pixel size exaggerated by 3x3 for better visualization .....	42
Figure 3.19a-g. Images showing observations at 07:30 hrs. In yellow the epicentre for the Changureh-Avaj earthquake (36,31). Pixels in green count 30-49% of the absolute maximum anomaly count for the whole time series. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%. Dashed yellow line, delineates part of the coast for the Caspian sea.....	44
Figure 3.20a-b. Observations taken from Saraf et al. (2008) reported results for the Changureh-Avaj earthquake using LST maps. The epicentre of the earthquake is represented by a star in both images. Areas with higher temperature are coloured in red.....	45
Figure 3.21a-g. Images showing observations at 12:30 hrs. In yellow the epicentre for the Firozabad-Kajoor earthquake (78,20). Pixels in magenta count 50-79%of the absolute maximum anomaly count for the whole time series. Pixels in green count 30-49%. Pixels in orange count 25-29%. Pixels in cyan count the 10-24%. Dashed yellow line, delineates part of the coast for the Caspian sea. ....	47

# LIST OF TABLES

---

Table 1-1. Seismic events where the presence of anomalies was previously reported by Saraf et al. (2008). 8

Table 1-2. List of typical Thermal Infrared (TIR) sensors used for land and sea surface temperatures and their characteristics. (Taken from Kuenzer et al., 2013).....10

Table 2-1. Characteristics of datasets for ordering; .....16

# 1. INTRODUCTION

Earthquakes; one of the most devastating natural hazardous events from which we are exposed to suffer any time in different zones all around the world. Knowing that, their prediction has become a topic of scientific interest. The mechanisms and processes behind the build-up of an earthquake are so complex, that so far give a sceptic overview of the topic. While some scientist claim the prediction of an earthquake as impossible, others report the observation of phenomenon able to warn for future seismic activity, the so called “earthquake precursors”. This is the case of anomalies observed in the remotely sensed thermal infrared (TIR) emissions which have been documented since Gorny et al., 1988 (Saraf et al., 2008). Some of the research suggesting as well the connection between anomalous TIR emissions and the presence of seismic activity studied earthquakes in e.g. China, Greece, India, Iran, Italy, Japan, Mexico, Russia and Turkey (Tronin et al., 2002, Saraf and Choudhury, 2005, Tramutoli et al., 2005, Ouzounov et al., 2006, Yao and Qiang, 2012). They reported the occurrence of the anomalies from few hours up to more than ten years before the main shock and spatially located at distances from several hundred up to thousand kilometres from the epicentre. There are two type of processes that could explain the presence of thermal anomalies due to seismic activity: atmospheric processes and Land Surface Temperature (LST) rise (Saraf et al., 2009). Some of the mechanism involving such processes are: (a) increase in air ionization due to changes in the air humidity and temperature (Pulinets et al., 2006), (b) light emissions due to positive holes charge during rock deformation by seismic activity, and (c) combined effects of soil moisture and gas concentration (Salman et al., 1992). Although, several limitations and short comings have been identified in the documented methodologies when reporting TIR anomalous emissions prior seismic activity. Sometimes, a proper definition of anomaly is missing, the meteorological effects in the images are not accounted for defining an anomaly and the amount of data use for observation is restricted, which makes a statistical interpretation difficult.

For that matter, a methodology which assess the definition of anomaly in the TIR emissions has been recently developed (Khan, 2010, Buzzo, 2012, Pavlidou, 2013). Such methodology, monitors the remotely sensed TIR emissions of a study area using long time series and its analysis is statistically supported. This research aimed to verify in an unbiased way, the existence of previously documented anomalies in the TIR emissions of ten different earthquakes occurred in Iran (

Table 1-1. Seismic events where the presence of anomalies was previously reported by Saraf et al. (2008).). Such anomalies were reported by Saraf et al. (2008), however, several limitations and short comings (i.e. missing observations due to cloud coverage or meteorological effect in the images) are identifiable in the methodology. In order to overcome such limitations, this research applied the existing methodology (Khan, 2010, Buzzo, 2012, Pavlidou, 2013) aiming to prove the reliability of using a longer time-series for observation and a cloud masking technic which attempts to remove the cloud cover in the TIR datasets (Khan, 2010, Buzzo, 2012) and helps to maintain the length of the time series for observation.

Saraf et al. (2008) claimed the findings after a visual analysis of LST maps made out of NOAA-AVHRR observations. They reported the appearance of anomalous dual and single peaks prior four moderate (>5Mw) and six strong (>6Mw) seismic events respectively. According to Saraf et al. (2008), “a detailed analysis was done to know the approximate time of appearance of a thermal anomaly (in terms of days), intensity of thermal rise and its spatial extent”. However, when reporting anomalous peaks in the datasets, the results leave questions regarding the spatial aspects (i.e. the extent size and location) of the anomaly appearance. Saraf et al. (2008) based their analysis in short time series using NOAA-AVHRR datasets, which sensor provides 4 day and night time daily images per satellite, those with 1.1km resolution. The

observations were limited to around a fortnight before and a fortnight after each event and there is a gap of LST maps in the time series. Even though NOAA-AVHRR have been documented to be useful for anomaly detection in the TIR prior seismic activity (Tronin et al., 2002, Saraf and Choudhury, 2005, Ouzounov et al., 2006), results to be in disadvantage with other sensors in the matter of monitoring TIR emissions as earthquakes precursors, such is the case of the first and second generation of Meteosat (MFG and MSG). Knowing that, this research was carried out using an existing methodology which was developed to work with such type of data (Khan, 2010, Buzzo, 2012, Pavlidou, 2013). The present study used Meteosat-5 datasets. Such geostationary satellite, provides images with a 30 minutes time gap. This allows a larger time series for monitoring fluctuations in the TIR. In the case of Saraf et al. (2008), the used time series was not equal for all the events. In some of the reported cases, the time series got shorter due to missing observations. Some of them were useless due to cloud cover, either the day of the main shock (i.e. Kerman earthquake) or over the “anomalous region” (i.e. Changureh-Avaj earthquake). In order to overcome this limitation, the existing methodology uses a cloud masking technique which helps to maintain the length of the time series and increase the statistical support of the analysis.

Table 1-1. Seismic events where the presence of anomalies was previously reported by Saraf et al. (2008).

Event no.	Earthquake	Origin		Location		Magnitude $M_w$ USGS	Depth Km
		Date	Time UTC	Lat° N	Long° E		
1	Changureh-Avaj	Jun 22 <sup>nd</sup> , 2002	02:58	35.63	49.05	6.5	10
2	Jahron	Jul 10 <sup>th</sup> , 2003	17:40	28.35	54.17	5.8	10
3	Kerman	Aug 21 <sup>st</sup> , 2003	04:02	29.05	59.77	5.9	20
4	Bam	Dec 26 <sup>th</sup> , 2003	01:56	29.00	58.34	6.6	10
5	Firozabad-Kajoor	May 28 <sup>th</sup> , 2004	12:38	36.29	52.59	6.3	28
6	Zarand	Feb 22 <sup>nd</sup> , 2005	02:25	30.75	56.82	6.4	14
7	Qeshm	Nov 27 <sup>th</sup> , 2005	10:22	26.77	55.86	6.0	10
8	Faryab	Feb 28 <sup>th</sup> , 2006	07:31	28.12	56.87	6.0	18
9	Fin	Mar 25 <sup>th</sup> , 2006	07:28	27.57	55.69	5.9	18
10	Persian Gulf	Jun 28 <sup>th</sup> , 2006	21:02	26.82	55.90	5.8	10

### 1.1. Remotely sensed thermal infrared data

The earth's surface temperature can be estimated with Thermal remote sensing measurements. Such technique, allows to determinate the amount of emitted electromagnetic radiation due to heat on the surface (emissions in the infrared region of the spectrum from 3-5 $\mu$ m and 8-14 $\mu$ m are measured). For that reason, thermal remote sensing does not depend on sunlight and can be done either day or night time. However, this technique is limited by the atmospheric absorption due to moisture content in the atmosphere. For that, the imagery should be calibrated according the sensor specifications, no matter what application the data will get.

Table 1-2, list the most used sensors for land and sea surface temperatures and related applications (Kuenzer et al., 2013).

Exist a large variety of satellites acquiring thermal infrared imagery for different applications i.e. climatology and evaporation, land surface temperature retrieval (LST) and earthquake precursors to name some. TIR imagery from FY-2 (Yao and Qiang, 2012) , NOAA-AVHRR (Saraf and Choudhury, 2005), MODIS (Saraf et al., 2009) to name some, have been documented to be useful for earthquake precursors analysis. Also, recent studies have been documented the use of the first and the second generation of Meteosat (MFG and MSG respectively) for study TIR emissions prior seismic activity (Khan, 2010, Buzzo, 2012, Pavlidou, 2013). This due to MFG and MSG geostationary orbits and hyper temporal resolution providing images every 30 and 15 minutes respectively.



Table 1-2. List of typical Thermal Infrared (TIR) sensors used for land and sea surface temperatures and their characteristics. (Taken from Kuenzer et al., 2013)

Sensor	Spatial resolution	Revisit	Swath width	Platform/satellite	Agency	Launch year
ETM+	60 m	16 d	185 km	Landsat 7	USGS, NASA	1999
TM	120 m	16 d	185 km	Landsat 5		1984
TIRS	90 m	16 d	185 km	Landsat 8 (LDCM)		2013
ASTER	160 m	4-16 d	60 km	Terra	NASA	1999
IRMSS	80 m	26 d	120 km	CBERS-1, 2, 2b	CRESDA, INPE	1999-2003, 2003, 2007-2010
IRSCAM	250 m	26 d	120 km	CBERS-3 and 4, 4b		2012, 2014, 2016
MERSI	300 m	1 d	2800 km	FY-3A, FY-3B	NRSCC, CAST, NSM a.o.	2008, 2010
InfraredCam	351 m	31 d	720 km	HJ-1B	CRESDA, CAST, NRSCC	2008
NIRST	370 m	<1-2 d	182-1,060 km	Auarius	NASA, CONAE	2011
BIRD	356 m	10 d	190 km	BIRD	DLR	2001-2004
TET-1	1.6 km	10 d	180 km	TET-1		2012
VIIRS	14 km	<1 d	3,000 km	Suomi NPP	NASA/NOAA	2011
CrIS	20 km	<1 d	2,200 km			2011
CERES	1 km	<1 d	3,00 km			2011
IIR	1 km	16 d	64 km	CALIPSO	CNES	2006
MODIS	1 km	4 per day	2,330 km	Terra, Aqua	NASA	1999, 2002
ATSR-2	1 km	3 d	512 km	ERS-2	ESA, UKSA, CSIRO	1995-2011
AATSR	1 km	35 d	500 km	Envisat	ESA, UKSA	2002-2012
AVHRR/1	1.1 km	<1 d	2,600 km	TIROS-N, NOAA 6, 8, 10	NOAA	1978-1986
AVHRR/2	1.1 km	<1 d	3,000 km	NOAA 9, 10, 11, 12, 13, 14		1984-2005
AVHRR/3	1.1 km	<1 d	3,000 km	NOAA 15-19, Metop A, B	NOAA, AUMETSAT	1998, 2000, 2002, 2005, 2006, 2012
MSG/SEVIRI	1-3 km	<1 d	Full earth disk	Meteosat-8/9/10	ESA/EUMETSAT	2002, 2005, 2012
MVIRI	5 km	<1 d	Full earth disk	Meteosat-3/4/5/6/7	EUMETSAT, ESA	1988, 1989, 1991, 1993, 1997
MSU-MR	1 km	Ev. 30 min	3,000 km	Meteor 3 M and -MN1	ROSHYDROMET a. o.	2001, 2009
MSU-GS	4 km	37 d	Full earth disk	Elektro-L N1	ROSHYDROMET a. o.	2011
IMAGER	4 km	Geost. <1d	Full earth disk	MTSAT-1, 2, 3	JMA	1999, 2006, 2013
MVISR	1.1 km	Geost. <1d	3,200 km	FY-1C, 1D	CMA, NRSCC	1999, 2002
IVISSR	5 km	3-4 d	Full earth disk	FY-2C, 2D, 2E, 2F	NRSCC, CAST, NSMC	2004, 2006, 2008, 2012
VHRR	8 km	1 d	Full earth disk	Insat-2A, B, E, -3A	ISRO	1992, 1993, 1999, 2003
IASI	25 km	Ev. 30 min	2,052 km	MetOP-A/B	CNES, AUMETSAT	2006, 2012
HIRS/3	20.3	29 d	2,240 km	NOAA 15, 16, 17	NOAA	1998, 2000, 2002
HIRS/4	20.3 km	<1 d	2,240 km	NOAA 18/19, MetOp A/B	NOAA, EUMETSAT	2005, 2006, 2009, 2012
ScaRab	40 km	1 d	2,200 km	Meteor-3, resurs-01, Megha Tropiques	CNES	1994, 1998, 2011

## 1.2. Existing methodology

The existing methodology first put forward by Khan (2010) and further developed by Buzzo (2012), was carried out using a package named “SHAKY” (previously developed by ITC team). Such package consists in a sequence of programs coded in IDL environment. Following, the description of the programs in order of execution.

### 1.2.1. Import

RAW data contains pixels values (DN) of thermal infrared captures. To convert DN to brightness temperature (BT) values in Kelvin, a sub-routine from the “SHAKY” package was used. Such program reads RAW images (.TIF files) and check for non-valid images (i.e. images with different dimension) and missing captions in the datasets. The sub-routine not only convert to BT values, but also calibrates the images for possible measuring errors or atmospheric absorption by moisture content. Calibration is important since wrong radiance values can change the statistics of the observations and by consequence distort the results. The calibration is done according the sensor specifications and calibration files are accessible through the EUMETSAT web portal<sup>1</sup>. The sub-routine makes a stack of the images and produces a Band Sequential (.bsq) image file than can be read using ENVI<sup>2</sup> software.

## 1.3. Cloud masking

An important fact to take in consideration when dealing with thermal imagery, is the coerture of clouds in the images. Cloud detection is important since the thermal infrared emission does not penetrate them. From there, areas with cloud coverage will show temperatures from the top of the cloud and not the actual temperature of the observed surface. Knowing this, images that contain pixels covered by clouds won't be useful for observation, will influence the statistics of the analysis and distort the results.

### a Relative-Absolute Mask (RAM)

Khan (2010) recently developed an algorithm which takes into consideration the temperature variations and the extreme weather conditions, both expected to naturally occur in any study area. Such algorithm takes two steps approach;

- Daily temperature variation range (Relative mask).  
A dynamic threshold was used knowing the climatological variations and weather patterns possibly due to different geomorphological features within the area. From the statistics of the data, a cut-off is set by 95% of the maximum observed temperature value minus the natural variation within the study area, this cut-off removes all similar or lower values in the scene.
- Minimum recorded temperature (Absolute mask).  
The lowest recorded temperature taken from the historical records for the study area is set as static threshold, masking all temperatures below that value as cloud covered.

After the Relative-Absolute Mask (Khan, 2010), the data still contains the effect of light cloud coverage. To overcome this limitation, some options have been developed, such is the case of Buzzo (2012) which tested a polynomial-fit (PF) mask assuming that removes cloud coverage from the scene.

### b Polynomial Fit (PF) mask

By definition, a polynomial fit regression models the relationship between two variables. Buzzo (2012) applied a polynomial regression in a year of MFG data. According the PF mask description, a yearly trend curve of temperature values can be described with a fourth degree polynomial model. The model then calculates a minimum threshold to discard low values due to e.g. clouds. A fit is calculated on a year of

---

<sup>1</sup> EUMETSAT Data Center. <http://www.eumetsat.int/website/home/Data/Products/Calibration/index.html>

<sup>2</sup> Environment for Visualizing Images (ENVI). <http://www.exelisvis.com/>

pixel values fluctuations. All values below the mean of the differences between the fit and the values lower than the fit, are masked as clouds.

Buzzo (2012) documented the performance of the PF mask applied to the imported data and after applying the RAMask (Khan, 2010).

### **1.3.2. Anomaly detection**

#### **a Pixel normalization**

After masking the clouds, there is still the need to neutralize the (a) meteorological and the (b) surface/subsurface effect in the images. In the matter of point (a) the effect is due to different weather conditions according e.g. different topography, different geomorphological settings. In the case of (b), the composition of the surface or the subsurface strongly influences the BT values for pixels for example on a coastal line due to wetness. Knowing that, a pixel normalization technique inherited from previous researches (Khan, 2010, Pavlidou, 2013) was applied to the data. A square ring of pixels is set by the user in order to divide each pixel value over the average value of the square ring. The size of the square ring must to be carefully chosen in the assumption that is close enough that all pixels inside have similar weather conditions. For example, when within the scene is encountered a coastal line, the pixel normalization could be a limitation for the pixels over the sea if the size of the square ring is too big. If that is the case, all pixels over the sea will be normalized with pixels from the land (which temperature is considerable unstable) and not with pixels from the sea (which temperature is more stable than the temperature in land).

#### **b Anomaly flagging**

After pixel normalization, the mean and the standard deviation of the statistical values are calculated for each pixel for the whole time series. A fixed threshold is set by the mean plus two standard deviations. Then, all pixels above the threshold are flagged as anomalous (Khan, 2010, Buzzo, 2012, Pavlidou, 2013). Following the anomaly flagging, an anomaly count is performed (Khan, 2010, Buzzo, 2012, Pavlidou, 2013). Knowing that an anomaly will probably appear and stay for few days prior the earthquake, the number of anomalous pixels in a time lapse window within the time series can be added up. (i.e. a number of days window will be moving through the entire time series, incident anomalies will be counted for multiple days.)

## 1.4. Research objectives

### 1.4.1. Main Objective

Apply time series analysis to the thermal infrared satellite data in order to verify the claimed appearance of anomalous peaks linked to seismic activity in ten different earthquakes.

### 1.4.2. Specific objectives

- i. Application of existing methodology to verify the presence of claimed thermal anomalies. Apply the methodology earlier proposed by Khan (2010) and further developed by Pavlidou (2013) in order to overcome the limitations found in other researches (i.e. short time series, no correction for weather conditions, no cloud masking, no hyper temporal analysis, no relation with the geographical location) and validate the shape, size and timing of anomalous peaks.
- ii. Test of existing methodology for different seismic event. Test the developed methodology (Pavlidou, 2013) on 10 earthquakes with different magnitudes, depths, focal mechanism and geomorphological features.

## 1.5. Research Questions

*About specific objective i;*

### Underlying processes

- Does addressing meteorological effects using a longer time-series gives a different view on the presence of thermal anomalies?

### Timing

- Where in time, anomalous peaks appear in the TIR datasets?
- How does a thermal anomaly look like in the TIR datasets? What is the difference between the appearance of single and dual peaks? When do they appear?
- What is the time extent of the anomaly appearance?

### Spatial aspects

- How do anomalies appear in space? In dependence of which factors:
  1. Depth,
  2. magnitude
  3. focal mechanism
  4. and geomorphological featuresanomalies are located and spatially extending?
- What is the spatial relation between the epicenter and the appearance of the anomaly?

*About specific objective ii;*

- Do we always observe thermal anomalies prior to earthquakes? For which factor:
  1. Depth,
  2. magnitude,
  3. focal mechanism,
  4. and geomorphological featuresdo we observe occurrence of thermal anomalies? What are the thresholds?

# 2. METHODOLOGY

The methodological steps followed for the assessment of the objectives of the present study are summarized in the flowchart below.

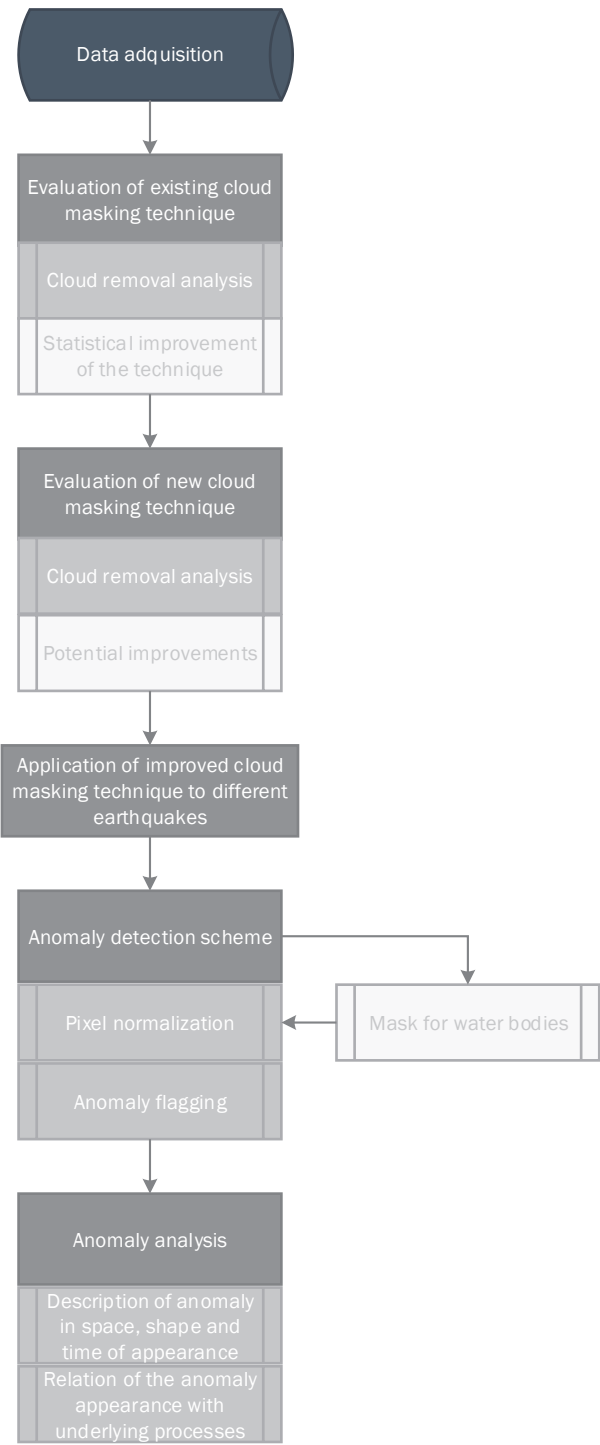


Figure 2.1. Flowchart of the methodological steps followed in the present study

The data processing was done in four steps;

- 1) Data acquisition;
- 2) Application of improved cloud masking technique to ten different earthquakes;
- 3) Anomaly detection;
- 4) Anomaly analysis.

A composition of subroutines running in IDL environment was used to perform the existent methodological steps (Khan, 2010, Buzzo, 2012). For exploring, visualization and manipulation of the data, ENVI software was used.

## 2.1. Datasets

This research was carried out using massive amounts of Meteosat-5 (MFG) data, knowing that, the expected time for processing was large. For ordering, the study area was divided in three different datasets. The extent of the datasets was done thinking of having enough pixels surrounding the earthquakes, this to perform the pixel normalization technique (see 1.3.2.a). Dataset 1 (shown in red in the image below) and dataset 2 (shown in green in the image below) are overlapping. This decision was made thinking about testing the performance of the existing methodology (see 0) and to compare results after processing datasets with different spatial extent. The length of the time series in years, was decided according the years where the earthquakes covered by each dataset occurred. For dataset 1 (See

Table 2-1) a total of 5 years of images with 1,600,539,600 points was used. For dataset 2 (See

Table 2-1), 4 years of data was used with 644,105,280 points. A total of 431,762,880 points from 4 years of data was used for dataset 3 (See

Table 2-1).



Figure 2.2. Delimitation and coverture of datasets used for the present study. Dataset 1, 2 and 3 are shown in red, green and yellow respectively. Google earth caption.



All data was ordered via the European Organization for the Exploitation of Meteorological Satellite (EUMETSAT) web portal<sup>3</sup> and received via email.

Basic information regarding general characteristics of each earthquake such as date, geographical location, depth and focal mechanism was obtained via the USGS<sup>4</sup> and the IIEES<sup>5</sup> web portals.

Table 2-1. Characteristics of datasets for ordering.

Dataset	Earthquakes covered	Time length (years)	Spatial extent (pixels)	Spatial extent (km)
1	Kerman	2002-2006	151x121	755x605
	Bam			
	Zarand			
2	Faryab	2003-2006	101x91	505x455
	Fin			
	Jahron			
	Queshm			
	Persian Gulf			
3	Changureh-Avaj	2001-2004	101x61	505x305
	Firozabad-Kajoor			

## 2.2. Import

The import of the images for each dataset, was carried out using the import program from the “SHAKY” package (see 1.2.1). Each dataset was processed separately, starting from dataset 1 (see 2.1) for time management and comparison of the results from previous works (see 0) with the present research. The import program was modified to perform the calibration correspondent to the years of observations for the Meteosat-5 (see 1.1). The product after running the import program, was a .bsq file stack of the images with BT converted values for each dataset. Such file can be read using ENVI software.

## 2.3. Cloud masking

As described in the Introduction of the present study, the coerture of clouds in the images results to be a limitation for a statistical analysis when monitoring for anomalies in the TIR emissions as earthquake precursors. For that, after the import of each dataset, the following program from the “SHAKY” package mask some pixels with low values as clouds (see 1.3.a and 1.3.b). After trying the cloud masking options from the existing methodology (see 1.3), the outputs showed light cloud coerture. To overcome such results, a modification of the algorithm was proposed for this study. (To see modifications of the code, go to Appendix)

### 2.3.1. Polynomial-Fit iterative method

After trying the PF mask as proposed by Buzzo (2012), some low values due to light cloud coverage were still recognised when exploring the data. For that reason, an iterative method of the PF mask was applied to the imported data. This approach was proposed aiming to remove the remaining clouds and increase the statistical relevance of the methodology.

<sup>3</sup> EUMETSAT Data Centre. <http://www.eumetsat.int/website/home/Data/DataDelivery/EUMETSATDataCentre/index.html>

<sup>4</sup> USGS National Earthquake Information Center. <http://earthquake.usgs.gov/earthquakes/eqarchives/epic/>

<sup>5</sup> International Institute of Earthquake Engineering and Seismology. [http://www.iiees.ac.ir/iiees/English/index\\_e.asp](http://www.iiees.ac.ir/iiees/English/index_e.asp)

### **2.3.2. Anomaly detection**

#### **a Pixel normalization**

A square ring of 12 pixels was set for applying the pixel normalization to the three datasets. For dataset 2 and 3 (see 2.1), pixels correspondent to the Persian gulf and the Caspian sea respectively were masked before applying the pixel normalization.

#### **b Anomaly flagging**

This methodological step is done with the last program in the sequence of the “SHAKY” package. As output, a single image for each dataset was created. Such image contains the maximum count of values flagged as anomalous within a 7 day timestamp (see 1.3.2.b). The timestamp works as a moving window 3.5 days centred. The 7 day timestamp was decided after the time reported for the anomaly appearance in previous works. (Saraf et al., 2008).



## 3. RESULTS

### 3.1. Methodology review and technical considerations

Working with such amount of data revealed the computational and time demand of the processes. Some technical problems such as insufficient memory for processing data were encountered during the different methodological steps.

Images showing the results from each step of the applied methodology will be shown in the following section. Also evidence of the corresponding improvements in the different methodological steps can be seen below.

#### 3.1.1. Import

The import of the images for each dataset was done sequentially, ordering images according to time of caption. Not a number (NaN) values were used in the stack for missing observations. The performance of the import for dataset 2 and 3 ran without technical issues. Although, for dataset 1 the import had to be done by year due to the large amount of data and insufficient memory for processing. Since the PFmask is based on a time series of a year and the pixel normalization works with a square ring in space on a single time slice (see 1.3.a and 1.3.2.a), the yearly import of dataset 1 result is irrelevant. However, anomaly flagging works with a moving window in time. For that, the five years of data for dataset 1, were merged after the pixel normalization.

#### 3.1.2. Polynomial-Fit iterative method

The following figures show a comparison of the PF mask (as proposed by Buzzo, 2012) and the Polynomial-fit iterative method.

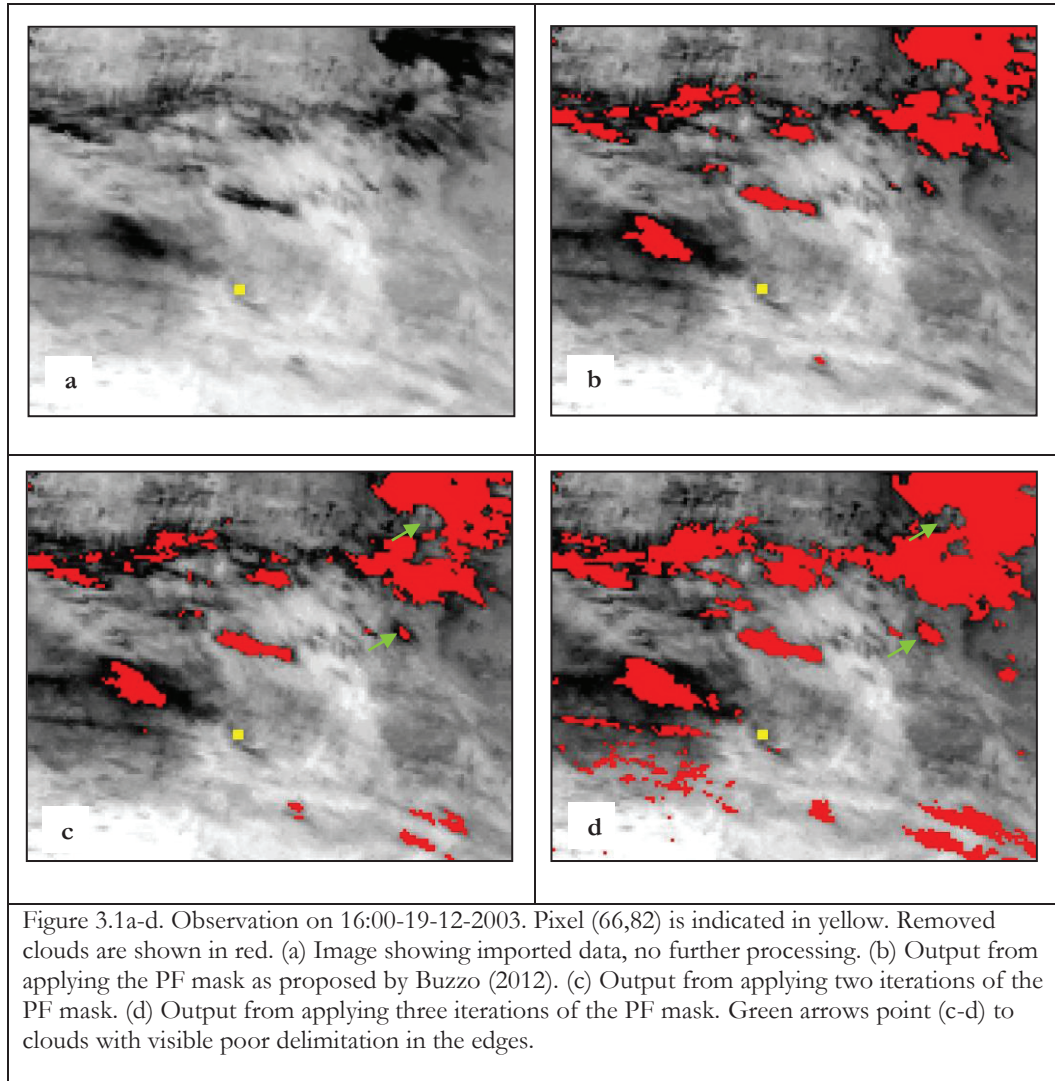
For comparison of results, pixel (66,82)<sup>6</sup> previously documented as anomalous (Khan, 2010, Buzzo, 2012, Pavlidou, 2013) is shown in the images below. For better visibility of the results, some time series graphs show two weeks subset of the whole time series. Images below show the imported data (See Figure 3.1a), the PF mask (See Figure 3.1b), as proposed by Buzzo (2012) and the performance of the Polynomial-Fit iterative method (See Figure 3.1c-d).

Observation from 16:00 hrs Dec 19 2003 was chosen to show results of the cloud masking technique. This due to the amount of cloud coverage in the images and being within the period where the findings for the Bam earthquake have been reported (Saraf et al., 2008).

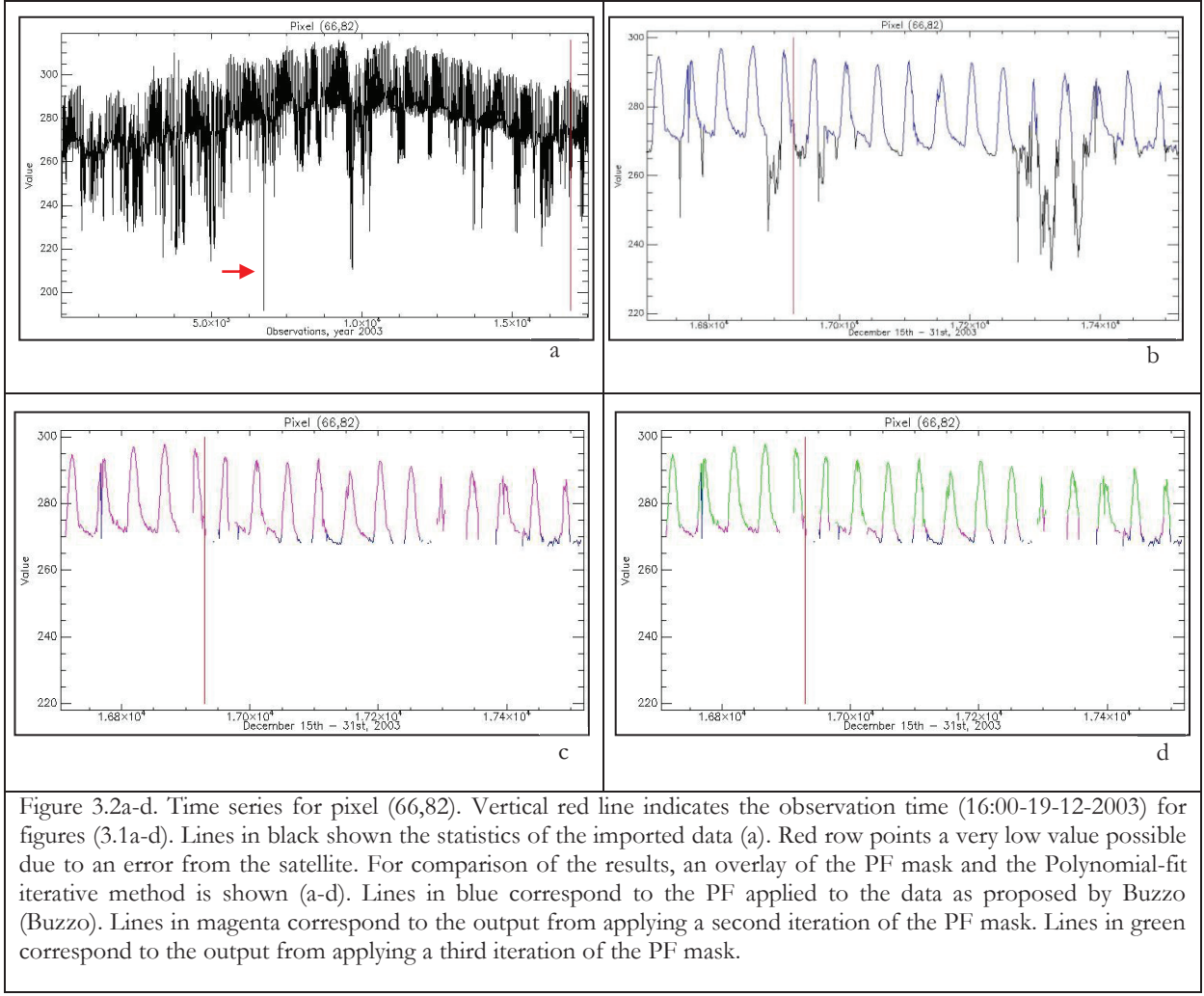
Two iterations of the PF mask are shown in figure 3.1c. From the caption it is possible to see the increment in the cloud removal. However, Figure 3.1c shows poor delineation of the edges of the clouds. This indicates that there was still an improvement needed. For that, a third iteration of the PF (figure 3.1d) was tested. It is possible to identify the increase in the cloud removal, however, from exploring the time series for pixel (66,82), a considerable loss of data was identifiable (i.e. mainly nights were removed). For the assessment of the results, the analysis of the polynomial-fit iterative method was based on time series plots (see Figure 3.2a-d). For supporting the analysis, a digital animation from the transit of masked clouds over the time series (i.e. animation of a series of images as in Figure 3.1a-d) was used.

---

<sup>6</sup> Different pixel coordinates from previous reports due to different extent of the dataset.



The following graphs show one year of radiant temperature for pixel (66,82). Low values are mainly due to light coverage in the images. In other cases, very low values are due to the measure of the top of a cloud (Figure 3.2a). For visualization and comparison of results after applying the polynomial-fit iterative method, following graphs show time slot for figure 3.1a-d.



After exploring the graphs, a second iteration of the PF mask resulted to suits better for the present study. In comparison with a third iteration of the PF mask (Figure 3.2b), a second iteration removes less data (night observations) when the polynomial curve is fitted (Figure 3.2a). Even though a coerture of light clouds will still remain in the images.

### 3.1.3. Pixel normalization

After applying two iterations of the PF mask (Polynomial-fit iterative method) to the data, the images still contain the meteorological and surface/subsurface effect (see 1.3.2.a). Figures below show the output graphs after applying the pixel normalization technique to the data. Looking at the findings previously reported (Pavlidou, 2013), a square ring of 12 pixels was used for this methodological step. Part of dataset 2 and dataset 3 contain pixels (see Figure 2.2) correspondent to the Persian Gulf and the Caspian sea respectively. For the matter of the surface/subsurface effect (see 1.3.2.a), pixels correspondent to the Persian Gulf and the Caspian sea were masked before the pixel normalization. The mask was build using ENVI software and based on the maximum and minimum pixel value for the coerture of sea in the images. The values were taken from a winter clouds-free night observation.

When looking at the outputs, some daily patterns were identified in some of the observations (time of appearance according the season). Such patterns are due to light morning fog, appearing between 07:00 and 09:00 hours in late winter observations (see Figure 3.3).

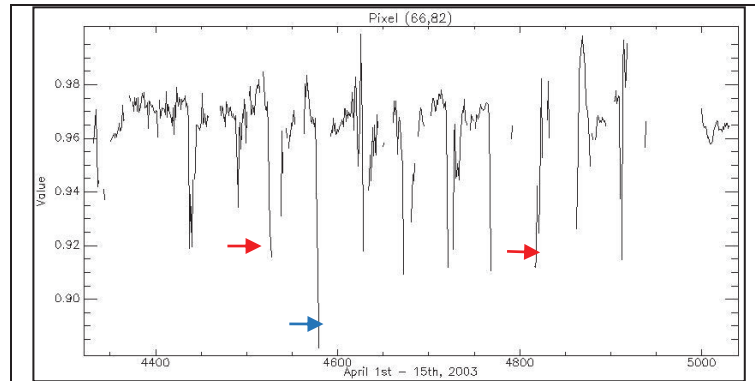


Figure 3.3. Statistics of pixel (66,82) after pixel normalization. Graph show daily morning patterns due to fog between 07:00 and 09:00 hrs. Red arrow pointing to a couple of the patterns. Blue arrow indicated a very low value due to the normalization of foggy/cloudy pixels.

#### 3.1.4. Anomaly flagging

Anomalies were flagged in each dataset and counted with a moving window of 7 days, all occurring anomalies were added up for the whole time series in each case.

For the present study, images corrected for missing values over the whole time series were used for the visual analysis of the observations and presenting results. For all figures, all discussed pixels and epicentres were oversized 3x3 for better visualization of the results.

#### a Dataset 1

Saraf et al. (2008) reported anomalies for the Kerman, Bam, Zarand, Faryab and Fin earthquakes using three different LST images with different extent each. To be consistent with the spatial extent of the area for observation surrounding each seismic event, one large dataset was used for the present study. For matter of visualization and spatial location, Figure 3.4 shows radiant temperature at the moment of the Bam earthquake. Epicentres of seismic events covered by the dataset were plotted on top for spatial location. In the figure, the seismic events are sequentially numbered according time of occurrence (e.g. 1. Kerman; 2. Bam; 3. Zarand; 4. Faryab; 5. Fin).

The absolute maximum count of anomalies, was 187 for the whole time series.

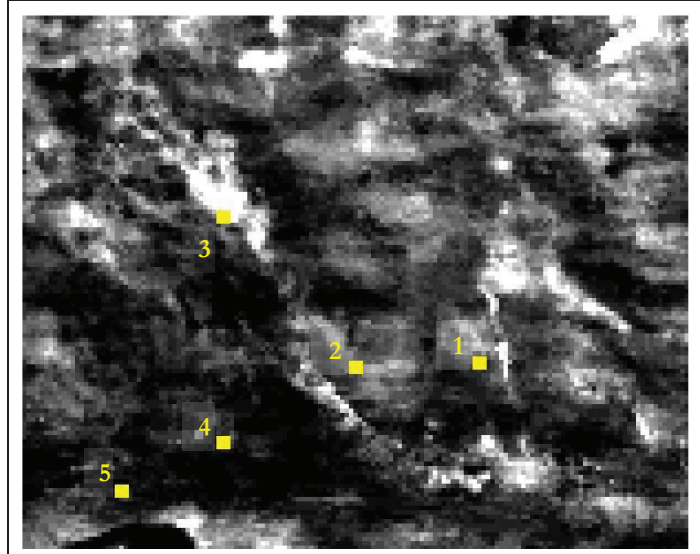


Figure 3.4. 02:00-26-12-2003 observation from dataset 1 (see 2.1) after correcting for missing values. Pixels in yellow correspond to earthquake epicentres: (1) Kerman; (2) Bam; (3) Zarand; (4) Faryab; (5) Fin. Pixel size exaggerated by 3x3 for better visualization.

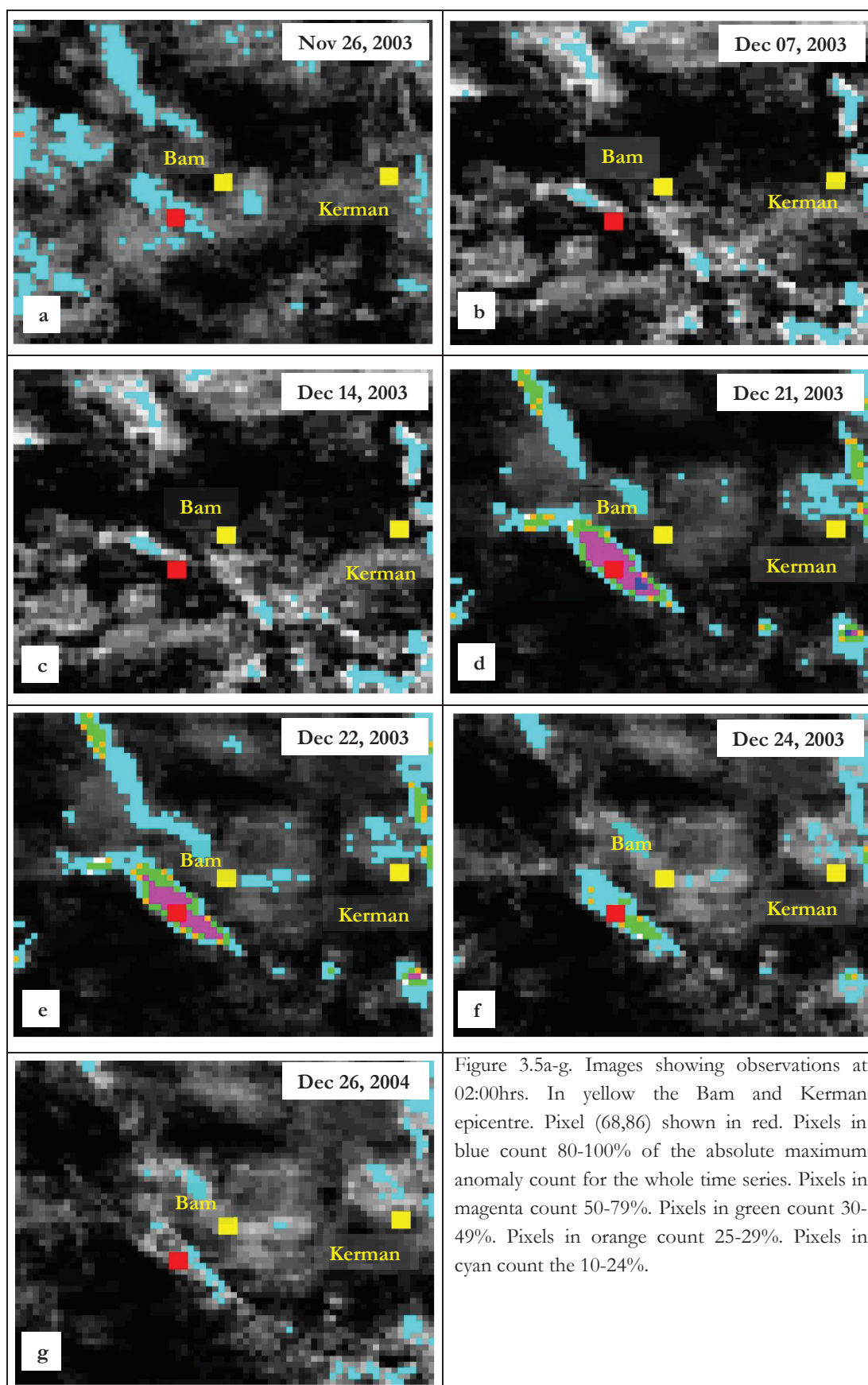
### Bam earthquake (01:56 UTC, December 26<sup>th</sup>, 2003).

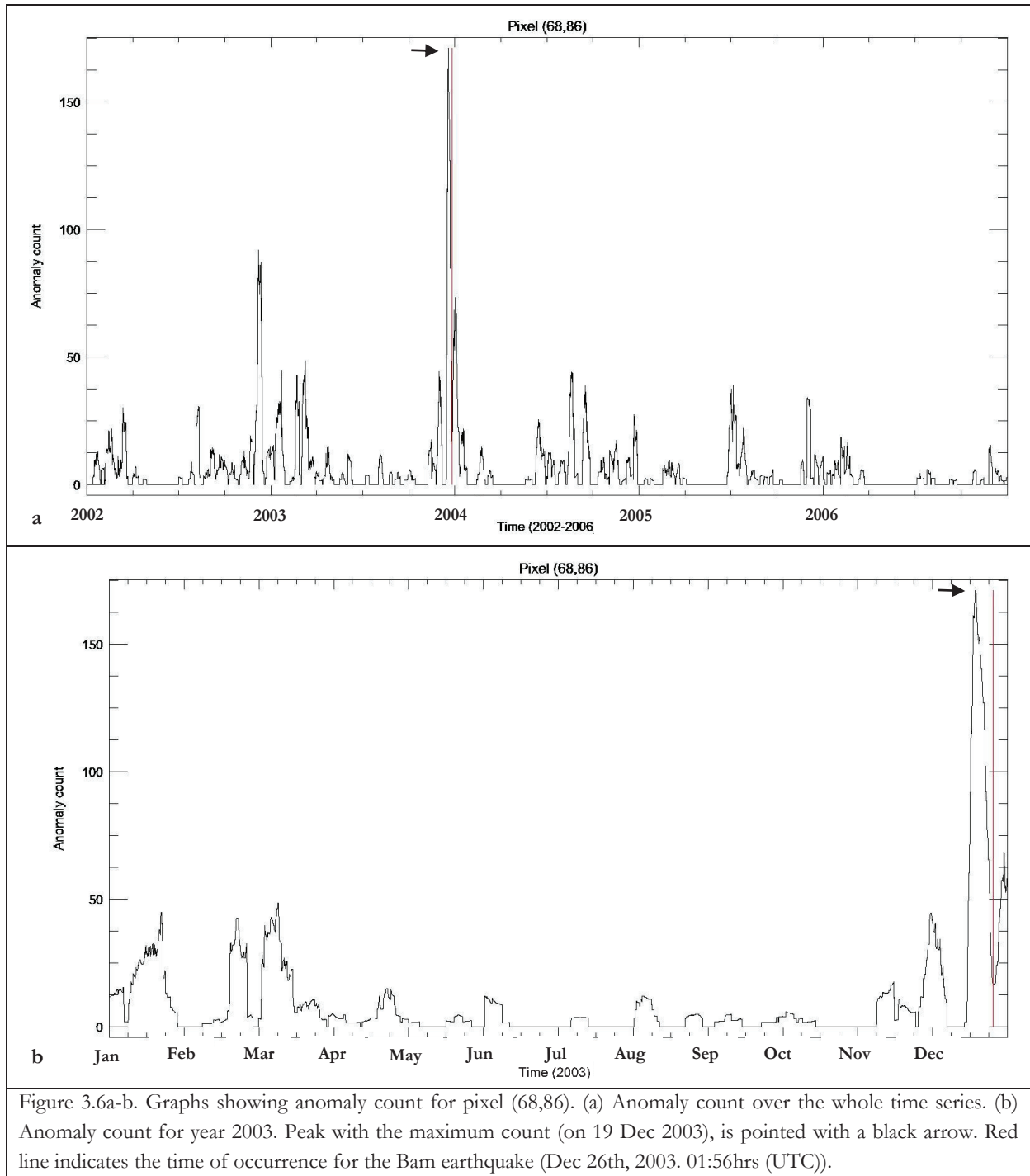
Saraf et al. (2008) reported a rise in the LST several days before the Bam earthquake. According to their observations “the first anomaly appeared on Dec 21<sup>st</sup>, 2003, the rise in temperature started on Dec 22<sup>nd</sup>, 2003 and attained the peak on Dec 24<sup>th</sup>, 2003”. Saraf et al. (2008) described a pattern extending around 30 km in N-S and to around 20km in the E-W direction from the Bam epicentre.

Figure 3.5a-g. show a time series subset of the results of the present study regarding the Bam earthquake. Figures show the observations from four weeks before and a week after the earthquake. The epicentre for the Kerman earthquake was plotted in the image due to its closeness in time and space to the Bam earthquake. Also pixel (68,86) was plotted in the image for having 160 anomalies count for the whole time series. The maximum count for pixel (68,86) for the whole time series, appears on 19 Dec 2003. Figure 3.5a-c show scattered areas with values between 18 and 45 counts on 26 Nov 2003, 07 Dec 2003 and 14 Dec 2003. By 21 Dec 2003, an area with higher count of flagged anomalies appeared (Figure 3.5d). On the observation for that day it is possible to identify an area with pixels counting between 93 and 149 flagged anomalies (e.g. between 50 and 79% of the absolute maximum count for the whole time series). The area is found at approximately 35km away from the Bam epicentre and has an approximate area of 1,200km<sup>2</sup>. A smaller area with values between 149 and 187 counts is also visible on the observation for that day. The small area is found at approximately 40km away from the epicentre with an approximate area of 25km<sup>2</sup>. On 22 Dec 2003 (Figure 3.5e), the area with 93-148 counts got smaller with approximately 90km<sup>2</sup>. Also the number of pixels with higher count disappeared. On 24 Dec 2004, the anomalous area is still visible, however with values between 18 and 46 counts for the majority of the contained pixels (Figure 3.5f). The area is found at approximately 30km away from the epicentre and with an approximate area of 1,900km<sup>2</sup>. By the time of the earthquake (Figure 3.5g), the high anomaly count exponentially decreased. In the observation for 26 Dec 2003, scattered areas with counts between 18 and 45 are identifiable. On observations after the earthquake, the low anomaly count remains on the observations (figures not shown in this paper)

Figure 3.6a-b show the graphs of the anomaly count for pixel (68,86). Figure 3.6a shows the anomaly count for the whole time series, there is possible to see that the highest count is 160. Figure 3.6b shows a year of the time series, there is more evident to visualize that pixel (68,86) attained its maximum count on 19 Dec 2003.

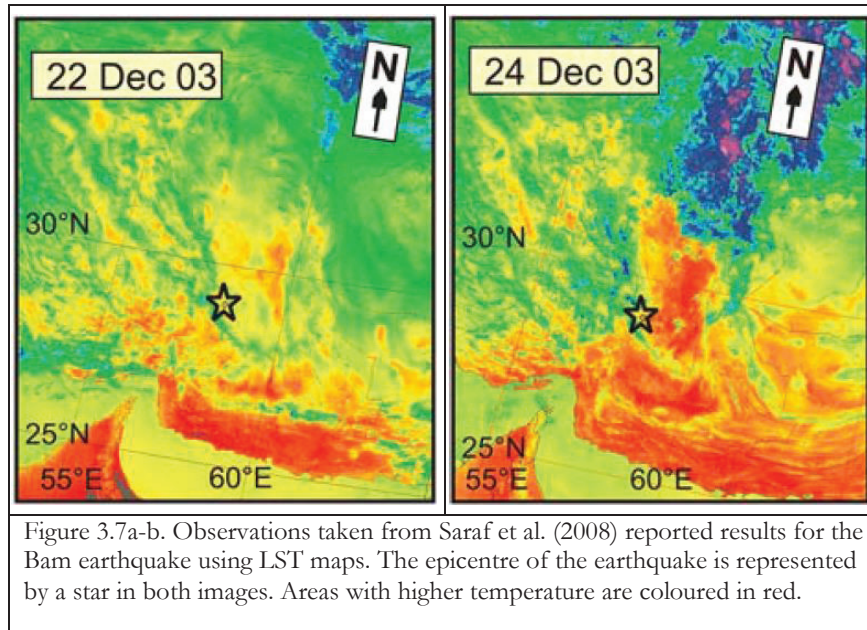








For the matter of comparison with previously reported findings, the following figures show results from Saraf et al. (2008). Figure 3.7 shows the observation from the day the anomalous area was reported to rise and attained its peak (Figure 3.7a and b respectively). The anomalous area is shown in red. However, no further description of the anomalous area in matter of covered area or distance from the epicentre was provided. From Figure 3.7a is possible to observe a very large area at a large distance from the epicentre. From Figure 3.7b, the anomalous area results even bigger and this time the area covers also the epicentre.



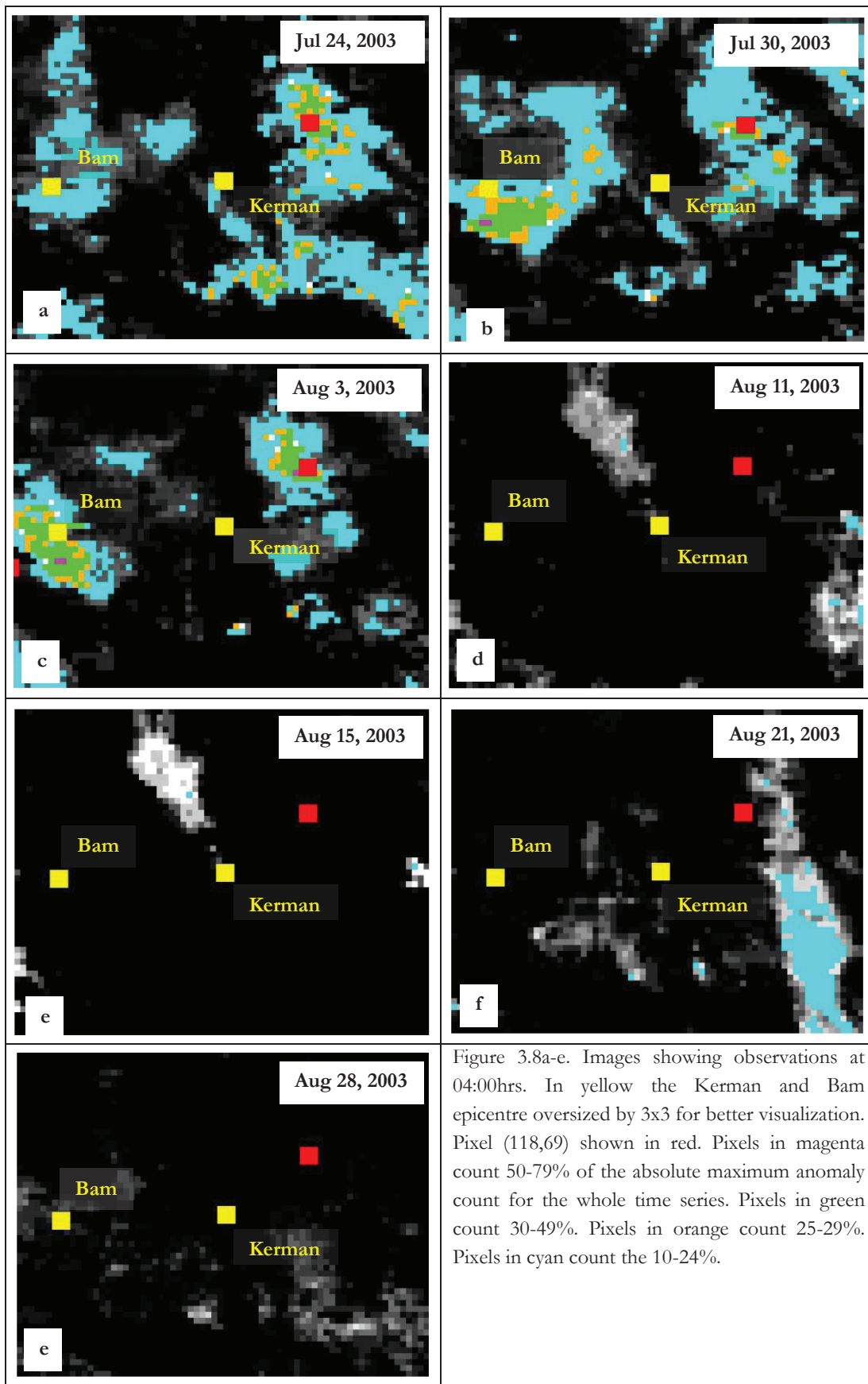
### Kerman earthquake (04:02 UTC, August 21<sup>st</sup>, 2003)

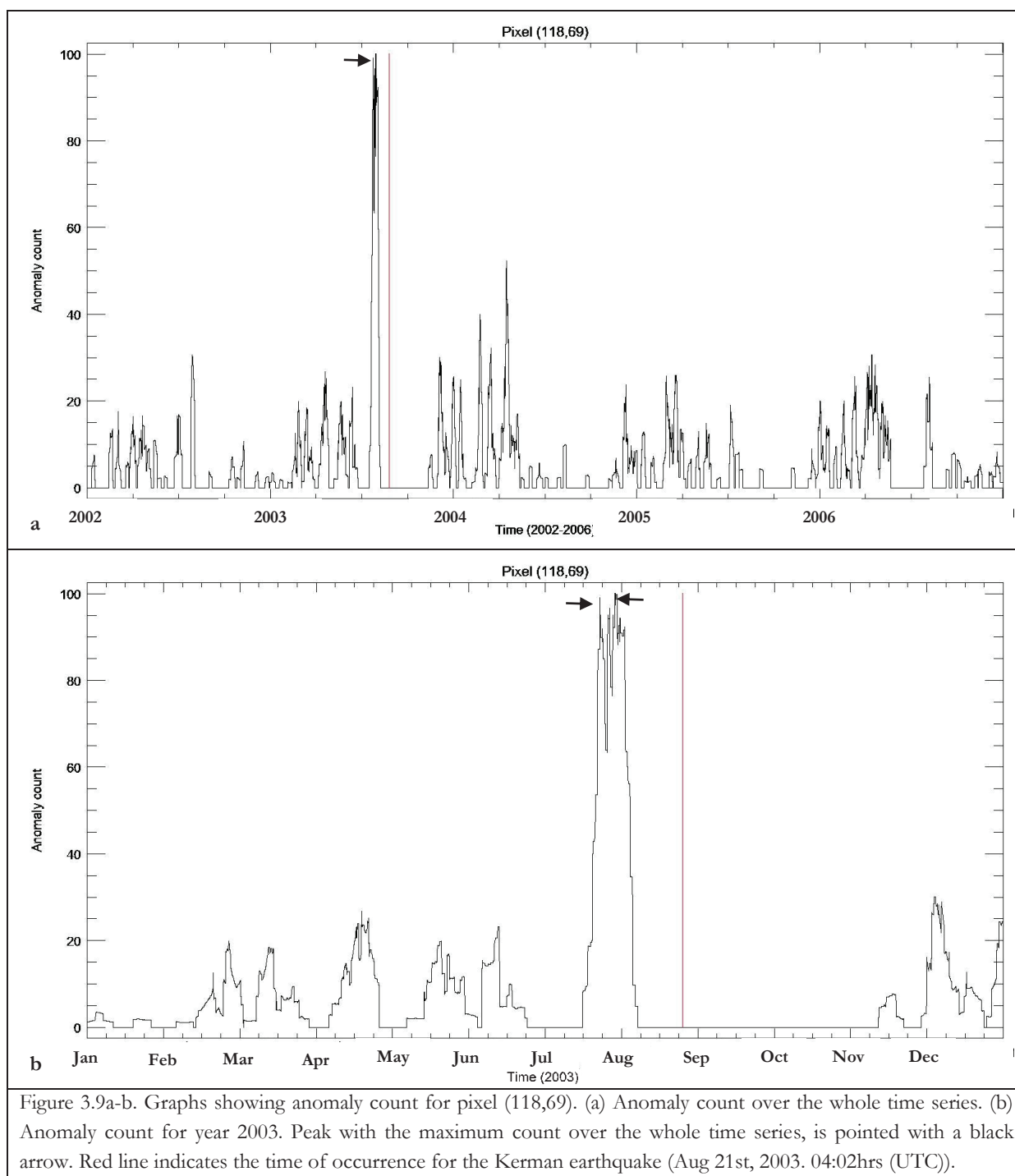
In the case of the Kerman earthquake, Saraf et al. (2008) reported a dual peak looking at a short time series. The first peak on 11 Aug 2003 and a second on 15 Aug 2003. Even though, LST maps were produced for those dates, no further information about the anomalous area on the maps was provided.

For the present study, Figure 3.8a-g show observations within a period of four weeks before and a week after the earthquake. For all figures, epicentres from the Bam and the Kerman earthquakes were plotted due to closeness in space and time of occurrence. Also pixel (118,69) was plotted on the images, such pixel shows its maximum counts over the whole time series starting on 24 Jul 2003 (a month before the earthquake).

On 24 Jul 2003 an area with anomaly count between 18 and 45 (e.g. 10-24% of the absolute maximum count for the whole time series) at approximate 70km away from the epicentre is found (Figure 3.8a). On the same observation, are found two smaller areas with counts between 46-55 (show in orange on the figure) and 56-92 (shown in green on the figure) respectively. Both areas get considerable reduced by 30 July 2003 (Figure 3.8b) and they increased again by 03 Aug 2003 (Figure 3.8c). By 11 Aug 2003 and 15 Aug 2003 not anomalous areas were identifiable on the observations (Figure 3.8d-e), e.g. all pixels in the area count less then 18 flagged anomalies which is less than the 10% of the absolute maximum value for the whole time series. By the time of the earthquake, an area with anomaly count between 18 and 45 appeared at approximately 110km distance away from the epicentre (Figure 3.8f).

Looking at the graphs of the anomaly count for pixel (118,69) over the whole time series, the figures show a peak with 95 counts on 24 Jul 2003 (Figure 3.9b), a month before the earthquake. The peak decreased, however still shows higher counts comparing counts for the whole time series (Figure 3.9a). On 30 Jul 2003 the peak attained its maximum of 99 counts over the whole time series. That is between the 50 and the 79% of the absolute maximum value for the whole time series.





For comparison with previously reported findings, following figures show results from Saraf et al. (2008). Figure 3.10a shows the observation from the day the dual peak was reported to appear for the first time. On 15 Aug 2003 (Figure 3.10b) was reported to appear again. However, no further information about the appearance of an anomalous area or its spatial location was provided. Coloured in red are the areas recognised by Saraf et al. (2008) with hottest temperatures and the epicentre is shown with a symbol star. For this case, also the lack of LST maps was reported due to cloud coverage in the images.

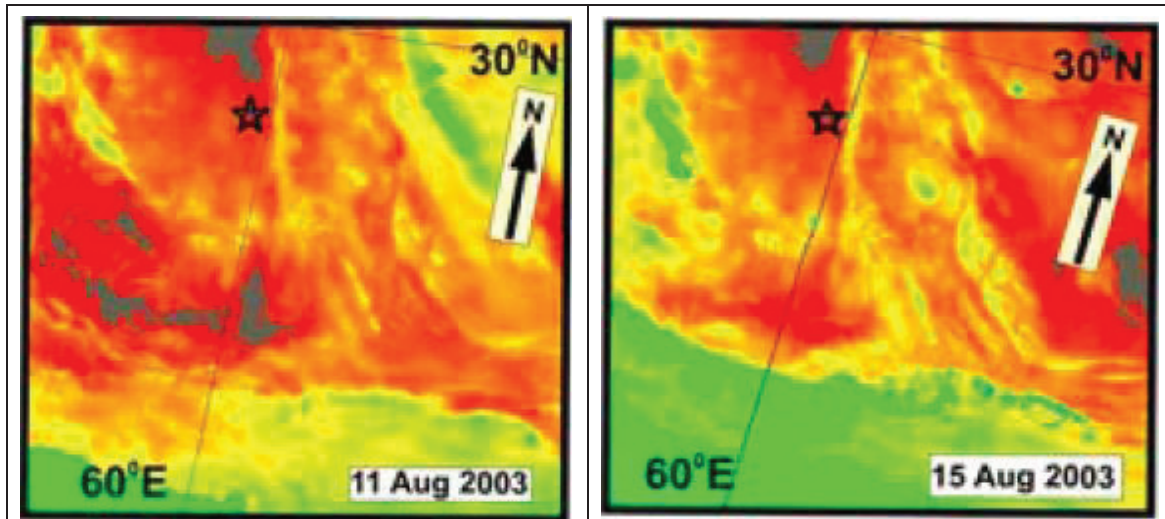


Figure 3.10a-b. Observations taken from Saraf et al. (2008) reported results for the Kerman earthquake using LST maps. The epicentre of the earthquake is represented by a star in both images. Areas with higher temperature are coloured in red.

### Zarand earthquake (02:25 UTC, February 22<sup>nd</sup>, 2005)

Saraf et al. (2008) reported an increasing anomalous area at about 75km diameter from the epicentre on 16 Feb 2005. Such area was reported to increase till 21 Feb 2005. However, no further description of the anomalous area or found peaks was provided.

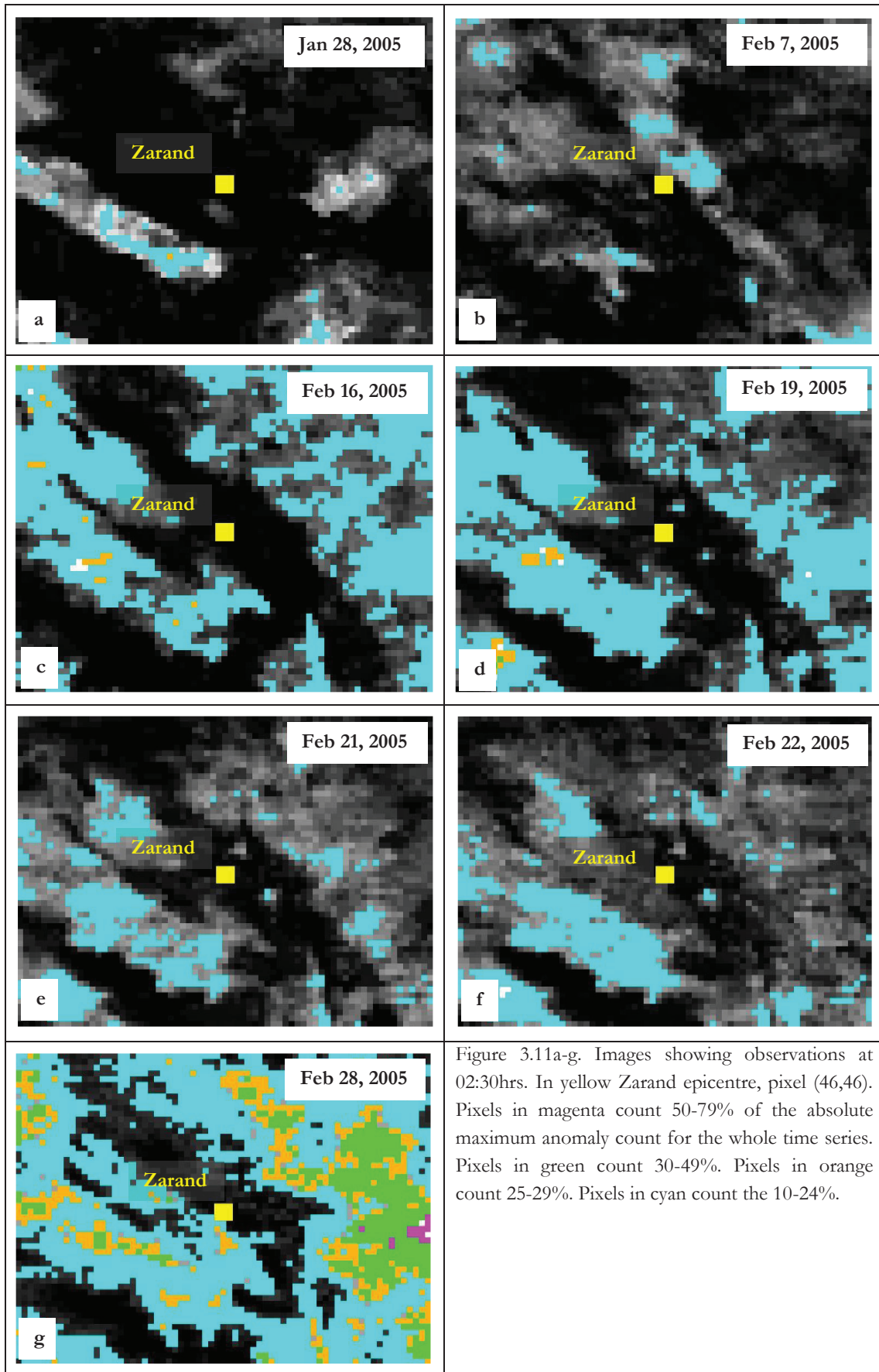
For the present study, a visual analysis of the observations and comparison with the graphs of the anomaly count for possible anomalous pixels near the epicentre was done. Following figures show observations from four weeks before and a week after the earthquake.

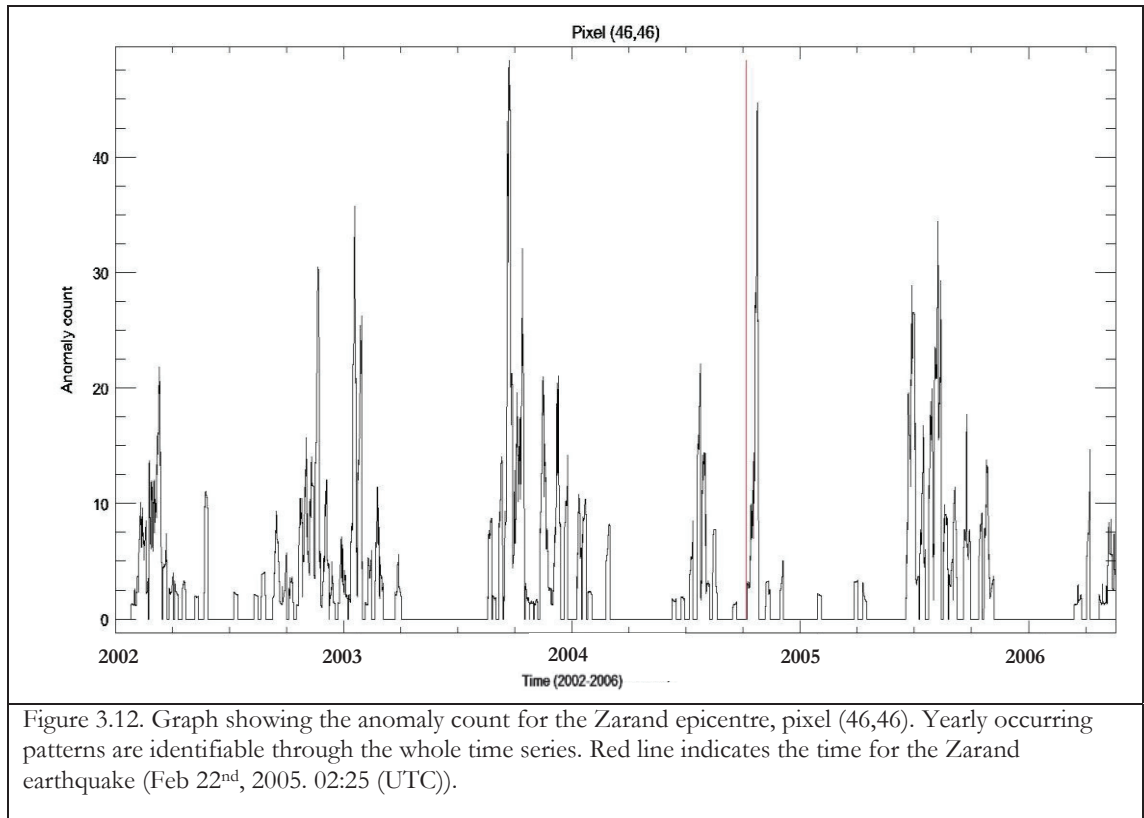
Figure 3.11a-b show observations from 28 Jan 2005 and 07 Feb 2005 respectively. In both images is possible to identify small areas with anomaly count between 18 and 45 flagged anomalies. Such areas appeared at approximately 55 and 25km respectively distance away from the epicentre (areas coloured in cyan on the figure). In the period between 16 Feb 2005 and 21 Feb 2005, a clear increment on areas covered by pixels counting 10-24% of the absolute maximum anomaly count for the whole time series is identifiable (Figure 3.11c-e). Now the areas are found at approximately 80km diameter from the epicentre. By the day of the earthquake (Figure 3.11d), the areas got smaller but still remain at approximate 35km distance away from the epicentre. On 28 Feb 2005 (a week after the earthquake), an exponential increment of areas with anomaly count between 18 and 46 anomalies flagged is identifiable (Figure 3.11g). By this date, also areas with anomalies with 25-29% and 30-49% of the absolute maximum anomaly count for the whole time series are identifiable.

The visual analysis did not show anomalies possible linked to seismic activity, but normal patterns due to small rise in the temperature of the area were identifiable from the animated time series.

Also Figure 3.12 shows the anomaly count for pixel (46,46) over the whole time series. For the case of the Zarand earthquake, pixel with anomalies possible link to seismic activity were no identifiable.

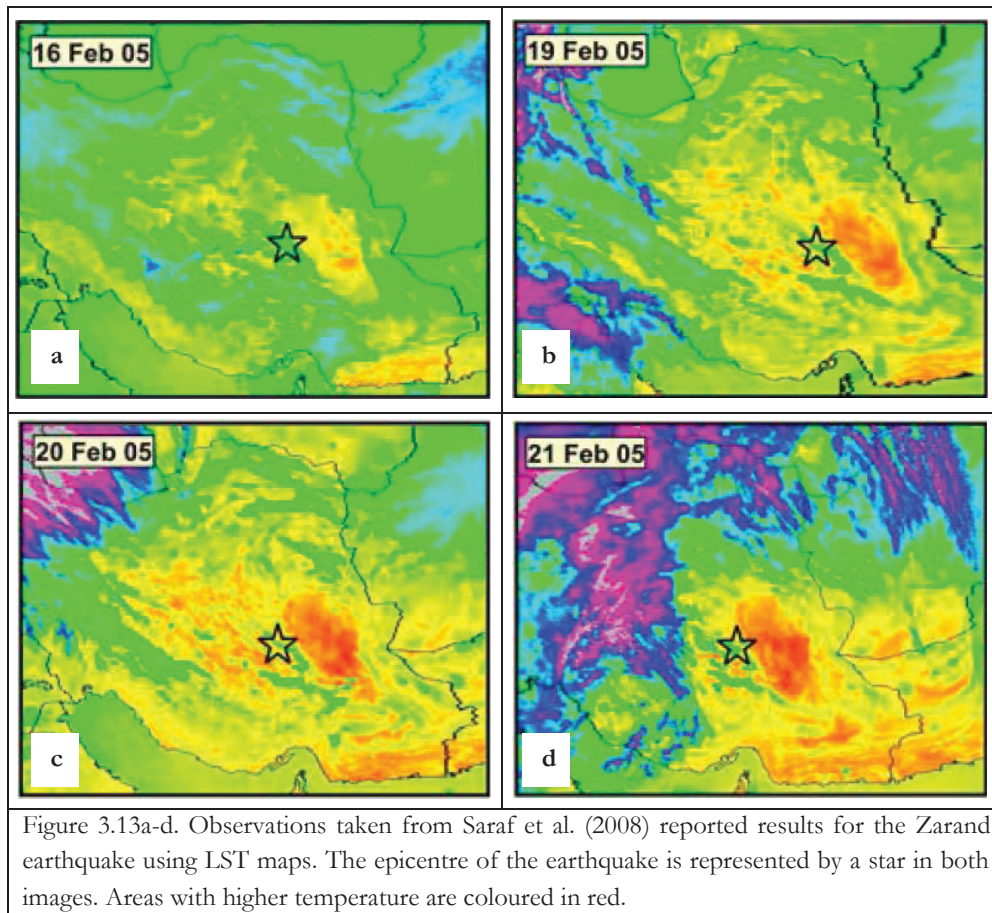








For comparison with previously reported finding in the matter of the Zarand earthquake, the following figures show the findings reported by Saraf et al. (2008). From Figure 3.13a, is possible to identify a small area that seems to increase by 19 Feb 2005 (Figure 3.13b), 20 Feb 2005 (Figure 3.13c) and till 21 Feb 2005 (Figure 3.13d). However, no further description for the found anomalous area was provided.



## b Dataset 2

For dataset 2, pixels over the coastline were normalized with pixels of NaN values (i.e. masked pixels due to coverage of water bodies e.g. Persian Gulf). Pixels with anomaly count >100 are found near the Jahron, Qeshm and Persian Gulf earthquakes (see Table 1-1 and Figure 2.2), this can be explain by the normalization with the NaN values. Figure 3.14 shows observation at the time when the Jahron earthquake occurred. For matter of visualization and spatial location, epicentres for all earthquakes covered by dataset 2 (Table 1-1) were plotted on the figure.

For the present study, images already corrected for missing values were used for the visual analysis of the observations. The maximum anomaly count for dataset 2 for the whole time series is 157 flagged anomalies.

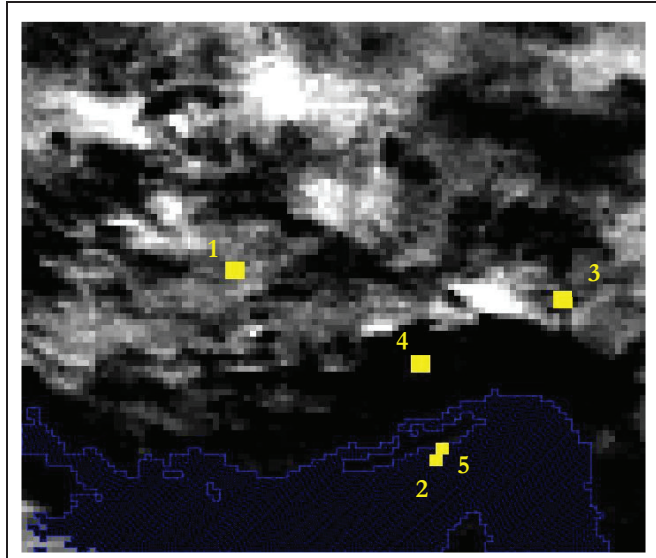
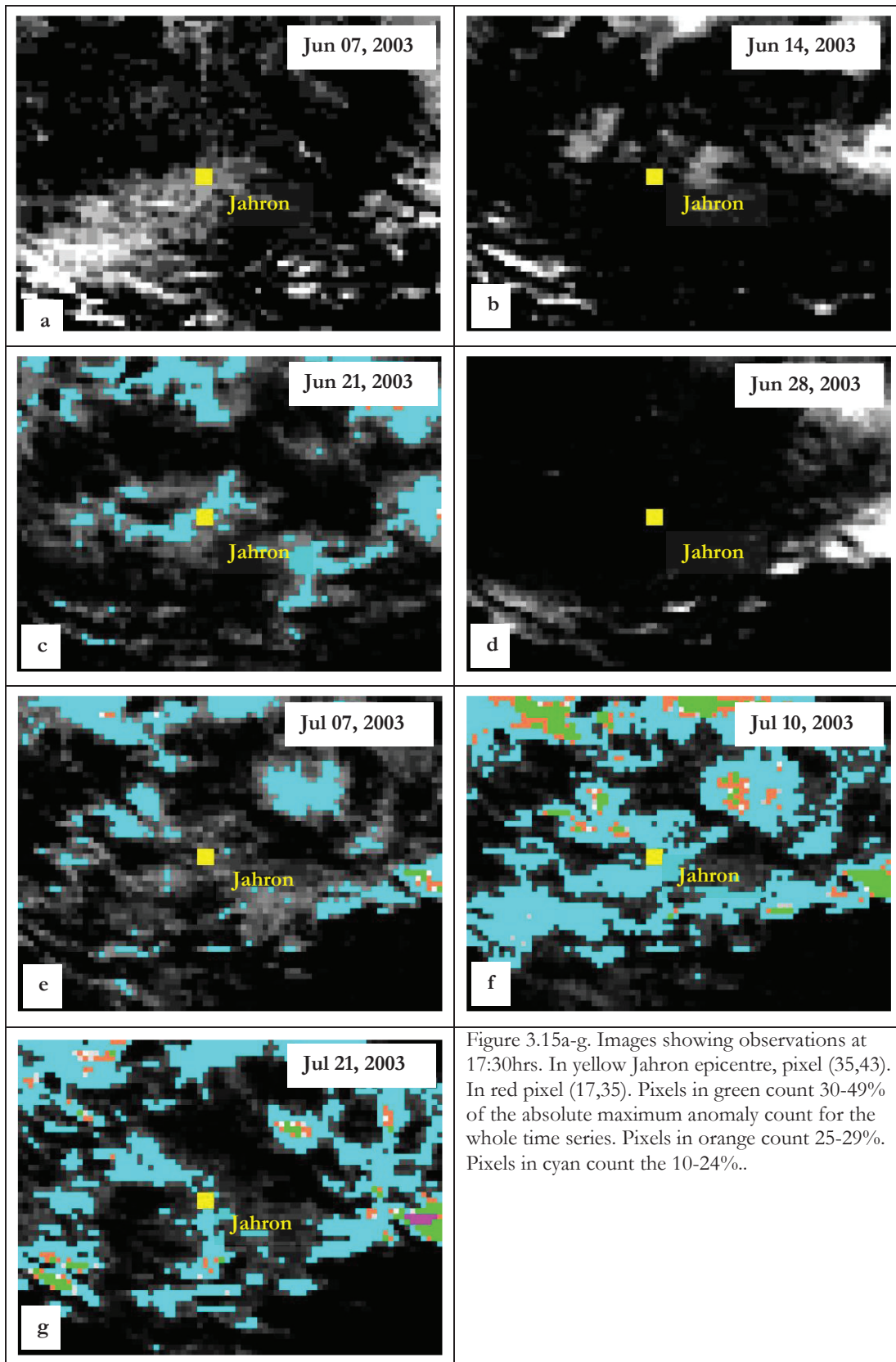


Figure 3.14. Image showing observation from dataset 2 when the Jahron earthquake occurred. Image already corrected for missing values for the whole time series. Pixels in yellow correspond to earthquake epicentres: (1) Jahron; (2) Qeshm; (3) Faryab; (4) Fin; (5) Persian Gulf. Pixel size exaggerated by 3x3 for better visualization

### Jahron (17:40 UTC, July 10<sup>th</sup>, 2003)

Saraf et al. (2008) reported a dual peak prior seismic events with magnitude bellow 6. In the case of the Jahron earthquake, the magnitude reported by the USGS was 5.8. For the present study and according Saraf et al. (2008) observations using a short time series, a dual peak was expected prior the earthquake but using a much longer time series. For the Jahron earthquake, Saraf et al. (2008) did not provide description of the findings.

Images bellow show observations from a month before and two weeks after the event. Observations for 7 and 14 June 2003 (Figure 3.15a-b) do not show anomaly count for pixels surrounding the Jahron earthquake. By 21 June 2003 (Figure 3.1c) scattered areas appeared with anomaly count between 15 and 38 flagged anomalies for the whole time series i.e. 10-24% of the absolute maximum anomaly count for the whole time series. One of the anomalous area covered the epicentre with an approximate area of 2,250 km<sup>2</sup>. For 28 Jun 2003 (Figure 3.1d) the anomalous area are not visible anymore. Again by 07 Jul 2003 (Figure 3.1e) scattered anomalous areas are found with 15-38 counts. The closest area is found at approximately 20km distance away from the epicenter. By 10 Jul 2003 (Figure 3.1f) areas with 15-38 counts potentially increased and small areas with 39-46 and 47-77 counts are identifiable near the epicentre. On the observation for 21 Jul 2003 all anomalous areas decreased in size. For that day, anomalous areas found near the epicentre are not visible anymore.



### Qeshm (10:22 UTC, November 27<sup>th</sup>, 2005)

Saraf et al. (2008) reported a single peak prior seismic events with magnitude equal or above 6. In the case of the Qeshm earthquake, the magnitude reported by the USGS was 6. For the Qeshm earthquake, Saraf et al. (2008) did not provide further description of their findings. For the present study and according Saraf et al. (2008) observations using a short time series, a single anomalous peak was expected prior the earthquake but using a much larger time series. For processing, the coverture of the Persian Gulf was previously masked using NaN values. When comparing with images corrected for missing values, all anomalous pixels recognized in images without correction, are not visible anymore. Since the visual analysis did not present anomalous observations, figures for the Qeshm are not following presented.

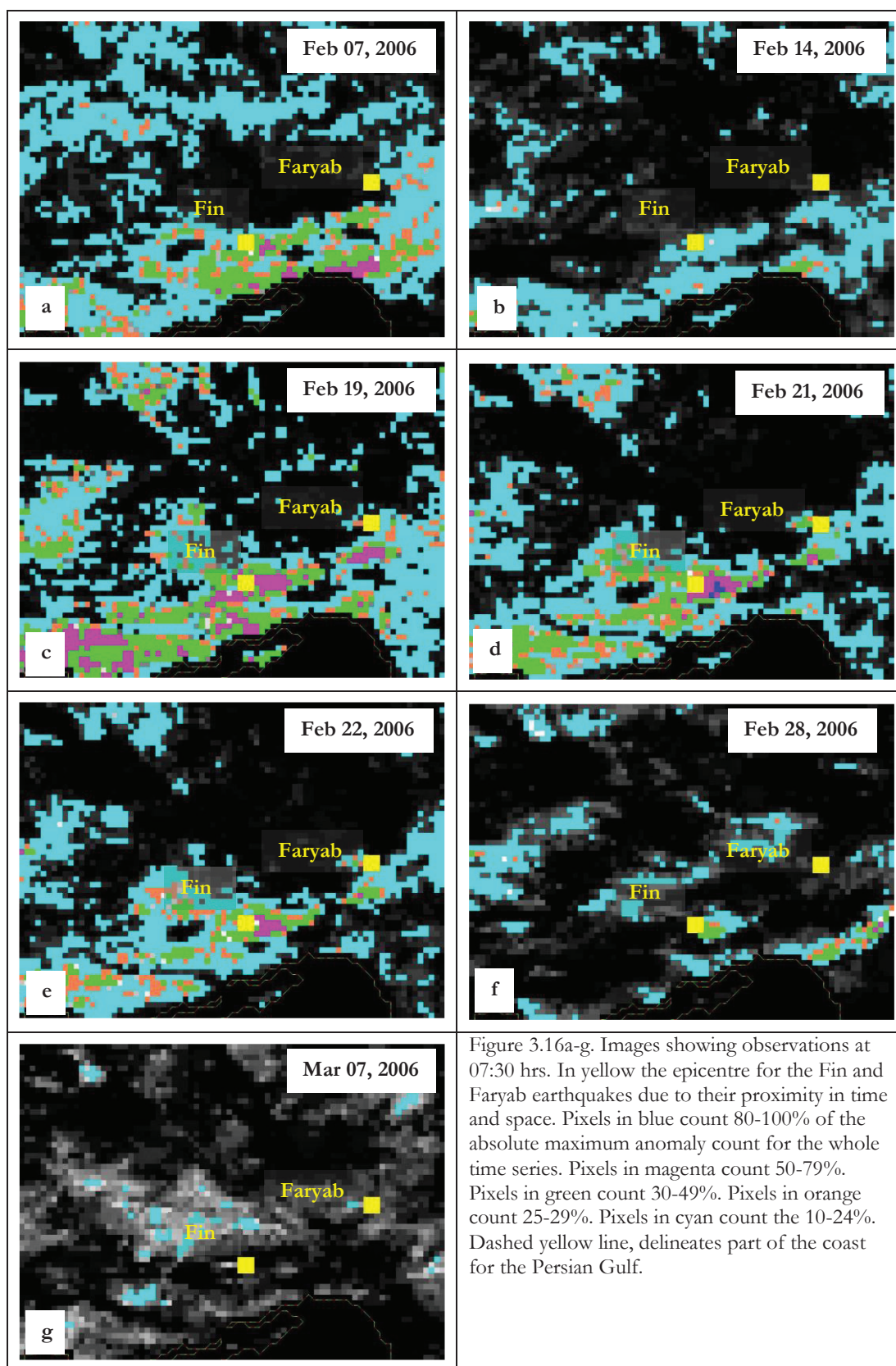
### Faryab (07:31 UTC, February 28<sup>th</sup>, 2006)

For the Faryab earthquake Saraf et al. (2008) did not provide description of the findings. However, they documented to find single peaks in the data for earthquakes with magnitude above 6. Such is the case of the Faryab earthquake with magnitude 6 reported by the USGS. For the present study, anomalous pixels were not identifiable for the Faryab earthquake.

Figure 3.16a-g. show observations from three weeks before and one week after the earthquake. On 07 Feb 2006 (Figure 3.16a) large areas with 15-38 anomaly count are identifiable surrounding the Faryab epicentre. Also small areas with 47-125 counts are seen neighbouring the coastal line of the Persian Gulf in the observation for 07 Feb 2006. On 14 Feb 2006 (Figure 3.16b) areas with 15-38 anomaly count remain on the observation but with a significant decrease on their extension. On 19-21 Feb 2006 (Figure 3.16c-e), areas with 47-125 anomaly count are recognizable again neighbouring the costal line of the Persian Gulf and surrounded by areas with 15-38 anomaly count. On 21 Feb 2006 (Figure 3.16e) a small area with count between 126-157 flagged anomalies is found. The area is located at approximately 15km away from the epicentre with an approximate area of 100km<sup>2</sup>. By the time of the earthquake (Figure 3.16f) all anomalous areas considerable decreased in size. However, areas with 39-77 are identifiable neighbouring the coastal line of the Persian Gulf. By 07 March 2006 (Figure 3.16g) very small areas with 15-38 anomaly count are identifiable surrounding the Fin earthquake which occurred on 25 March 2006. On the observation for that day, anomalous areas observed neighbouring the coastal line in previous days are not appearing anymore.

In the present study, pixels possible linked to seismic activity were not identifiable for the case of the Faryab earthquake.





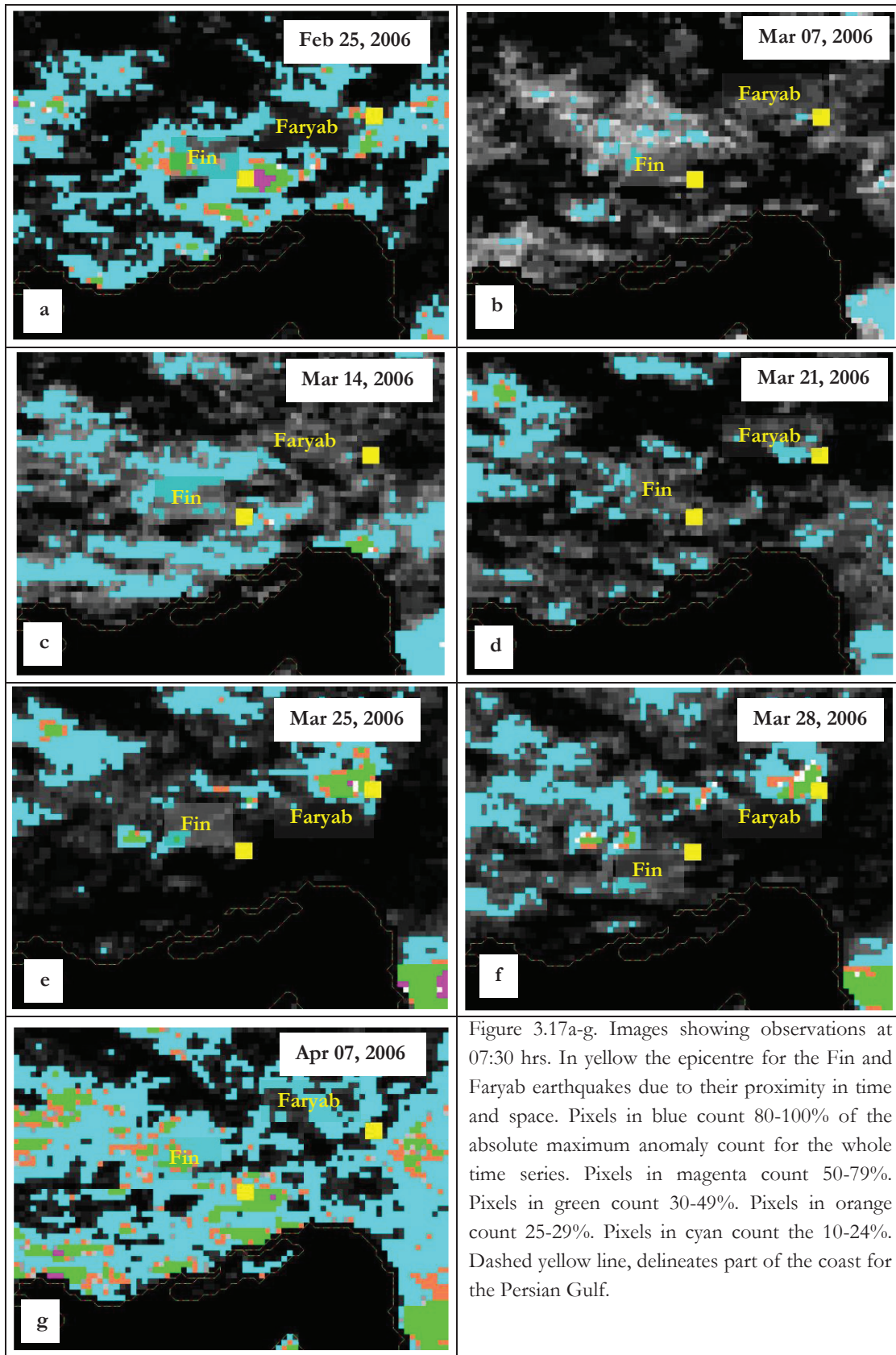
### Fin (07:28 UTC, March 25<sup>th</sup>, 2006)

For the Fin earthquake, Saraf et al. (2008) did not documented a description of their findings. The Fin earthquake was reported by the USGS with a magnitude 5.9. For the present study, results did not show anomalous areas. All anomaly counts were below 15 counts (e.g. <10% count of the absolute maximum value for the whole time series for the entire dataset).

Following figures show observations from a month before and two weeks after the earthquake.

On 25 Feb 2006 (Figure 3.17a), a large area is identifiable with values between 15 and 38 anomalies count for the whole time series. By 07 Mar 2006 (Figure 3.17b), the area considerable decreased in size. On 25 and 28 March 2006 (Figure 3.17c-d) scattered areas appear again with values between 15 and 38 anomalies count for the whole time series. By 07 Apr 2006 (Figure 3.17g), areas similar to the observed on 25 Feb 2006 are recognizable in the image.

In the present study, anomalous pixels or areas possible link to the presence of seismic activity, were not identify for the case of the Fin earthquake.





### Persian Gulf (21:02 UTC, June 28<sup>th</sup>, 2006)

Saraf et al. (2008) reported a dual anomalous peak prior seismic events with magnitude below 6. In the case of the Persian Gulf earthquake, the magnitude reported by the USGS was 5.8. For the present study and according Saraf et al. (2008) observations using a short time series, a dual anomalous peak was expected prior the earthquake but using a much larger time series. For the Persian Gul earthquake, Saraf et al. (2008) did not provide further description of their observations. For processing, the coerture of the Persian Gulf was previously masked using NaN values (Same case as the Qeshm earthquake). When comparing with images corrected for missing values, all anomalous pixels are not visible anymore. Since the results did not show anomalous areas or pixels possible to link to the Persian Gulf earthquake, figures showing the observations were excluded from the present study.

**c**

### **Dataset 3**

In dataset 3, a small north-east part of the dataset correspond to the Caspian Sea south coast. Before processing, all pixels correspondent to the Caspian sea were masked with NaN values.

For matter of visualization and spatial location, Figure 3.18 shows observation from the Changureh-Avaj earthquake. Epicentres for earthquakes covered by dataset where plotted on the figure for their spatial localization.

The maximum anomaly count in the image for the whole time series is 122 counts.

For the present study, images already corrected for missing values were used for the visual analysis of the observations.

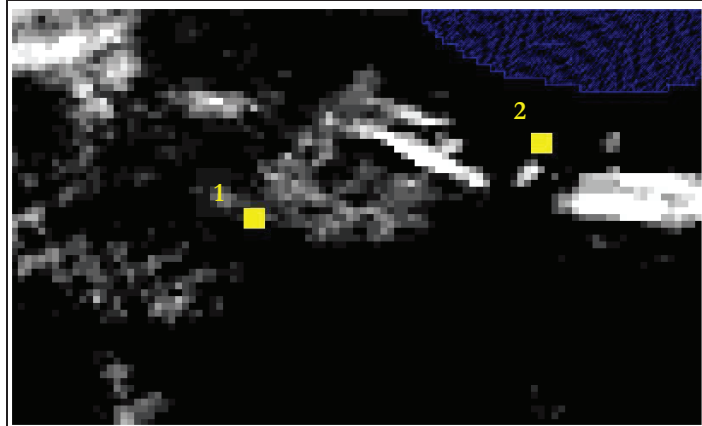


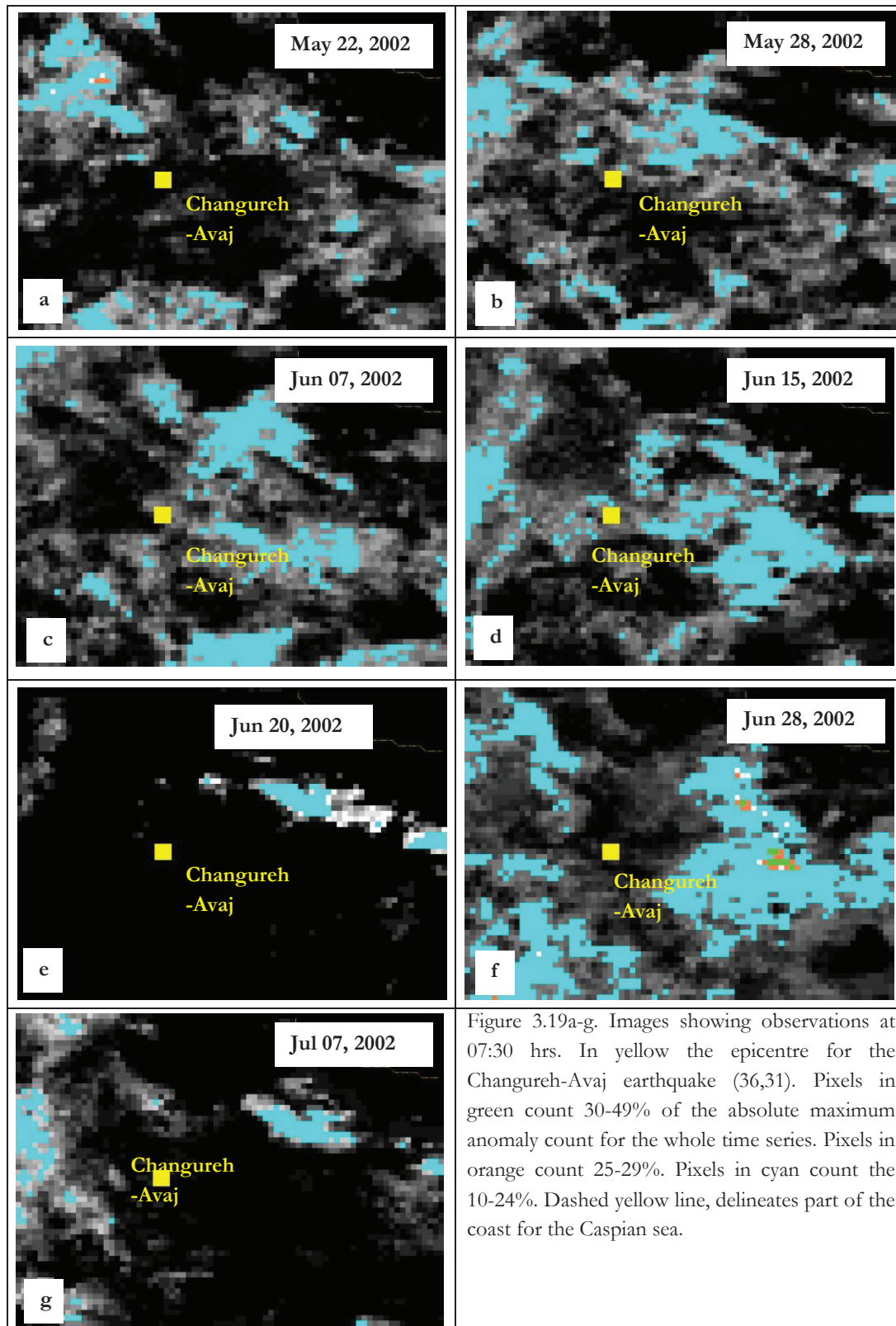
Figure 3.18. Image showing observation from dataset 3 when the Changureh-Avaj earthquake occurred. Image already corrected for missing values for the whole time series. Pixels in yellow correspond to earthquake epicentres: (1) Changureh-Avaj; (2) Firozabad-Kajoor. Pixel size exaggerated by 3x3 for better visualization

### Changureh-Avaj (02:58 UTC, June 22<sup>nd</sup>, 2002)

Saraf et al. (2008) documented an anomalous area that started developing on 15 Jun 2002 for the Changureh-Avaj earthquake. Such anomaly, reached its maximum on 20 Jun 2002. In the report, they described the extent of the anomalous area with 164,000 km<sup>2</sup>, however no further information was provided.

Following images show results from a month before and two weeks after the earthquake. Figure 3.19a-d show areas with anomaly count between 12 and 29 flagged anomalies in observations from 22 May 2002 to 15 Jun 2002. By 20 Jun 2002 (Figure 3.19e) the anomalous area exponentially decreased in size. The area is found at approximately 105km distance away from the epicentre with an approximate area of 975km<sup>2</sup>. By 28 Jun 2002 the area exponentially increased again (Figure 3.19f) and decreased again by 07 Jul 2002.

In the case of the Changureh-Avaj earthquake, anomalous areas or pixels suggesting future seismic activity were not found.



For comparison results previously reported by Saraf et al. (2008), following images show their reported findings. Figure 3.20a shows observation on 15 Jun 2002 from where the appearance of an anomalous area was reported by Saraf et al. (2008). Such anomalous area was reported to attain its maximum on 20 Jun 2002 (Figure 3.20b). The area was reported with 164,000km<sup>2</sup>, however no further information was provided. From Figure 3.20a-b, is possible to assume the describe area reported as anomalous is coloured in red.

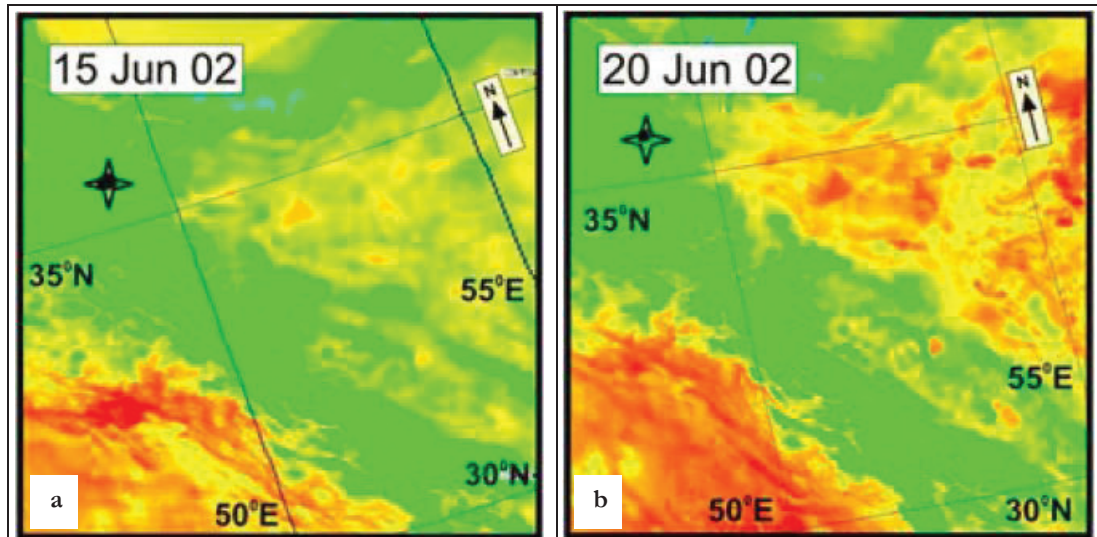


Figure 3.20a-b. Observations taken from Saraf et al. (2008) reported results for the Changureh-Avaj earthquake using LST maps. The epicentre of the earthquake is represented by a star in both images. Areas with higher temperature are coloured in red

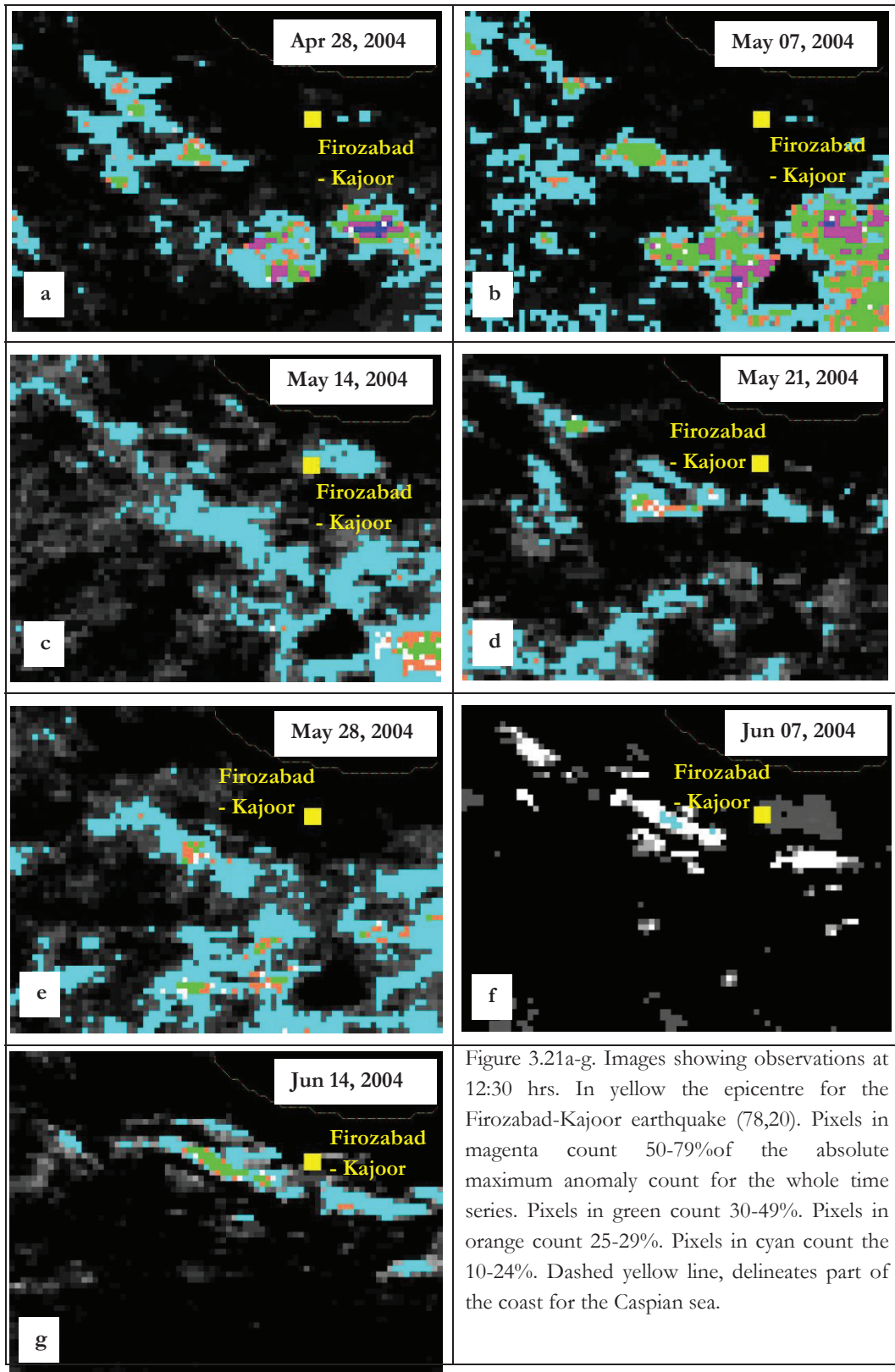
#### Firozabad-Kajoor (12:38 UTC, May 28th, 2004)

For the Firozabad-Kajoor earthquake, Saraf et al. (2008) did not describe any anomalous area in the observations. However, they did reported a single anomalous peak for seismic event with magnitude above 6. In the case of the Firozabad-Kajoor earthquake, the USGS reported a magnitude 6.3 for the event.

Following figures show observations from a month before and two weeks after the Firozabad-Kajoor earthquake. On 28 Apr 2004 and 27 May 2004, Figure 3.21a-b show spread areas with majority of pixels with count between 10 and 24% of the absolute maximum anomaly count for the whole time series. Smaller areas with 30-49%, 50-79% and 80-100% are also identifiable in the figures. By 14 May, 21 May and 28 May (Figure 3.21c-e) areas with 30-49% count still remain in majority. On 07 Jun 2004 (Figure 3.21f), a week after the earthquake, those areas exponentially decreased in size and by 14 Jun 2004 (Figure 3.21g) are visible again.

The results for the present study did not show any anomalous area in the observations or peak in the time series. Areas with high anomaly count were found far from the epicentre i.e. approximately 100km distance away.







## 4. DISCUSSION

The present study aimed to verify the presence of anomalies in the TIR emissions prior to ten earthquakes in Iran. These seismic events were previously documented for showing anomalies prior their occurrence. All studied seismic events have different characteristics (i.e. different magnitude, depth, time of occurrence, geomorphological settings) and were originally studied by Saraf et al. (2008) using different length of time series for observation. The visual analysis was carried out using LST maps made of NOAA-AVHRR imagery which provides 4 daily images. The use of NOAA-AVHRR datasets gives already a short time series for observation and the coerture of clouds and meteorological effect in the images make even shorter time series. For that matter, the present study was carried out using an existing methodology which applies a cloud masking technique based on the statistics of the datasets. Such methodology overcomes the meteorological and surface/subsurface effect in the data and affecting the results. Also, the existing methodology is applicable to MFG and MSG imagery. With this characteristics, the existing methodology allowed the present study to use and maintain a longer time series for observation i.e. 2,676,407,760 points were used for observation.

The key components of the present study were the different view of the anomalies after addressing (a) cloud coerture, (b) meteorological effect on the images and (c) the use of a longer time series for observation.

The review of the cloud masking technique, showed some important issues to consider for anomaly detection. For example, the existing technique underlined the presence of considerable amount of cloud coverage after it was applied to the data. However, even after the improvements of the technique, the cloud mask failed to detect all clouds edges and the data showed to still contain a light coerture of clouds. This is a very important factor to consider for further research since this light coerture of clouds affects the data after normalization. For that, future efforts should be focused on for example, growing region algorithm for the matter of a better delimitation of the clouds. Also, a challenging part will be to monitor the track of the light clouds and identify their pattern e.g. due to morning fog that will last for a couple of hours.

All study earthquakes were clustered in three different datasets. For the performance of the pixel normalization technique, the Fin and Faryab earthquakes were both clustered in dataset 1 and 2 (see Figure 2.2). This compare results from using datasets with different areas for observation for each earthquake. However after applying the normalization to the data, the results showed the effect on the edges of the images for both cases, for that, observations described in the results section are from using dataset 2 in both cases. For the case of the Fin earthquake, normalization with pixels covering water content (results for such observations are not described in the present study) showed areas with high anomaly count over the coastal line for the Persian Gulf. Such effects result to be important concerning the anomaly detection. No masking water content on the images affects the statistics in the matter of choosing thresholds for anomaly flagging. Then this will results in high anomaly count over a costal line.

Overall, the present study focused in the view of anomalies in the TIR emissions. Results showed that a very important factor to consider for anomaly detection, is the length of the time series for observation. This study maintained the length of the time series for observation. When comparing with findings previously made by Saraf et al. (2008), the use of a longer time series for observation in the present study, discard some of the previously reported anomalies. Saraf et al. (2008) just provided a description for the Changureh-Avaj, Kerman, Bam and Zarand earthquake. In the case of the Changureh-Avaj earthquake, results from the present study show that the use of a longer time series and the compensation for missing values give a better view to the presence of an anomaly. In this case, what can be look as an anomalous

area possible linked to seismic activity using a short time series, the use of a longer time series shows a normal occurring pattern.

For the case of the Kerman earthquake, Saraf et al. (2008) provided limited description about their findings. However when comparing with the results from the present study, both show different dates for the anomaly appearance. In results from the present study, the maximum anomaly appeared 24 Jul 2003 (a month before the earthquake). The peak decreased between 25 Jul and 29 Jul 2003 and increase again with a high anomaly count for 30 Jul 2003. This result is completely different from the one reported for Saraf et al. (2008) where they documented the appearance of a peak on 11 Aug 2003 which attained its maximum anomaly on 15 Aug 2003. For this case, is again evident the importance of using a longer time series which gives a more critical view of its presence.

For the Bam earthquake, Saraf et al. (2008) provided more information about their findings. However, in comparison with the results from the present study, the extent of the reported anomalies results to be smaller than the reported from Saraf et al. (2008) which from the images appears to be spreading over a very large area.

In the case of the Zarand earthquake, when comparing with results from the present study, the use of a longer time series results to be better for detection. Since looking at a larger amount of observations, findings that appear to be anomalous in a short period of time will be discard as anomalous comparing with more observations over time. This also will help the recognition of occurring patterns through a certain period of time.

In conclusion, the present study proved the importance of the length of the time series use for the analysis of the TIR emissions as earthquake precursors. Overall, the results showed to give a better view in space and time of the identify anomalies. Also, the present study showed the efficiency of addressing the meteorological effect and cloud coverage in the images, both influencing the results.

Furthermore, the present study lead to improvements in the performance of the existing methodology recognizing its actual limitations.

## LIST OF REFERENCES

---

- BUZZO, G. 2012. Earthquake precursors detected via hypertemporal satellite thermal data.: University of Twente Faculty of Geo-Information and Earth Observation (ITC).
- KHAN, N. A. 2010. *Thermal infrared time series analysis for earthquake precursory detection*. University of Twente Faculty of Geo-Information and Earth Observation (ITC).
- KUENZER, C., GUO, H., OTTINGER, M., ZHANG, J. & DECH, S. 2013. Spaceborne Thermal Infrared Observation – An Overview of Most Frequently Used Sensors for Applied Research. In: KUENZER, C. & DECH, S. (eds.) *Thermal Infrared Remote Sensing*. Springer Netherlands.
- OUZOUNOV, D., BRYANT, N., LOGAN, T., PULINET, S. & TAYLOR, P. 2006. Satellite thermal IR phenomena associated with some of the major earthquakes in 1999–2003. *Physics and Chemistry of the Earth, Parts A/B/C*, 31, 154-163.
- PAVLIDOU, E. 2013. *Time series analysis of remotely sensed TIR emission preceding strong earthquakes*. University of Twente Faculty of Geo-Information and Earth Observation (ITC).
- PULINET, S. A., OUZOUNOV, D., KARELIN, A. V., BOYARCHUK, K. A. & POKHMELNYKH, L. A. 2006. The physical nature of thermal anomalies observed before strong earthquakes. *Physics and Chemistry of the Earth, Parts A/B/C*, 31, 143-153.
- SALMAN, A. O., EGAN, W. G. & TRONIN, A. Y. Infrared remote sensing of seismic disturbances. 1992. 205-218.
- SARAF, A., RAWAT, V., BANERJEE, P., CHOUDHURY, S., PANDA, S., DASGUPTA, S. & DAS, J. D. 2008. Satellite detection of earthquake thermal infrared precursors in Iran. *Natural Hazards*, 47, 119-135.
- SARAF, A. K. & CHOUDHURY, S. 2005. Cover: NOAA-AVHRR detects thermal anomaly associated with the 26 January 2001 Bhuj earthquake, Gujarat, India. *International Journal of Remote Sensing*, 26, 1065-1073.
- SARAF, A. K., RAWAT, V., CHOUDHURY, S., DASGUPTA, S. & DAS, J. 2009. Advances in understanding of the mechanism for generation of earthquake thermal precursors detected by satellites. *International Journal of Applied Earth Observation and Geoinformation*, 11, 373-379.
- TRAMUTOLI, V., ALIANO, C., CORRADO, R., FILIZZOLA, C., GENZANO, N., LISI, M., MARTINELLI, G. & PERGOLA, N. 2013. On the possible origin of thermal infrared radiation (TIR) anomalies in earthquake-prone areas observed using robust satellite techniques (RST). *Chemical Geology*, 339, 157-168.
- TRAMUTOLI, V., CUOMO, V., FILIZZOLA, C., PERGOLA, N. & PIETRAPERTOSA, C. 2005. Assessing the potential of thermal infrared satellite surveys for monitoring seismically active areas: The case of Kocaeli (Izmit) earthquake, August 17, 1999. *Remote Sensing of Environment*, 96, 409-426.
- TRONIN, A. A., HAYAKAWA, M. & MOLCHANOV, O. A. 2002. Thermal IR satellite data application for earthquake research in Japan and China. *Journal of Geodynamics*, 33, 519-534.
- YAO, Q.-L. & QIANG, Z.-J. 2012. Thermal infrared anomalies as a precursor of strong earthquakes in the distant future. *Natural Hazards*, 62, 991-1003.

## APPENDIX I

---

### Shaky package

- Program for import  
pro complete\_import\_met5, ev  
  
compile\_opt idl2  
  
title = 'complete\_import\_met5'  
  
; select meteosat tiff files  
file\_dir=dialog\_pickfile(/read,/directory,/must\_exist)  
file\_names=file\_search(file\_dir,'\*.tif')  
  
;check error  
if (file\_names[0]) eq '' then begin  
print, 'error filename'  
return  
endif  
  
; query first input file  
ok = query\_tiff(file\_names[0],tif\_info,geotiff=geotiff)  
;help, geotiff  
if (Pulinets et al.) eq 0 then begin  
print, 'error tiff'  
return  
endif  
  
; define data dimensions  
ns = tif\_info.dimensions[0]  
nl = tif\_info.dimensions[1]  
nel = n\_elements(file\_names)  
  
; create array for bandnames (date strings)  
b\_names = strarr(Tramutoli et al.)  
  
;create arrays for all years and their day numbers, months and their day numbers, days, hours,  
minutes  
all\_years = lonarr(Tramutoli et al.)  
yr\_days=lonarr(Tramutoli et al.)  
mon = lonarr(Tramutoli et al.)  
m\_days=lonarr(Tramutoli et al.)  
dd = lonarr(Tramutoli et al.)  
hh = lonarr(Tramutoli et al.)  
minutes = lonarr(Tramutoli et al.)

```

;(A)create a b_names array with elements=names of files i, only as date, imported either from
mfg or from msg
for i = 0, nel-1 do begin
    file_name = file_names[i]
    ; add filename datestring to bandnames array
    file_name = file_basename(file_name,'.tif')
    ;if IR108 exists in whichever file, then make bandname array following mfg procedure, else
msg
    image_source='IR108'
    if STRCMP(image_source, file_name , 5 ) then begin
        b_names[i] = strmid(file_name,strpos(file_name,'-',/reverse_search)+1)
        hourly_images=2
    endif else begin
        b_names[i] = (strsplit(file_name,'_','/extract'))[0]+'00'
        hourly_images=4
    endelse
endfor
;sort the resulting b_names array
b_names=b_names[sort(b_names)]

```

```

;(B)split the name further to extract year, month, day, hour, minutes sorted arrays//create
sorted array of unique years
for j=0, nel-1 do begin
    b_name=b_names[j]
    ;extract year string
    year_string=strmid(b_name, 0, 4)
    ;convert year string to long and store it in all_years array
    all_years[j]= long(year_string)
    ;create all months array
    mon[j]=long(strmid(b_name, 4, 2))
    ;create all days array
    dd[j]=long(strmid(b_name, 6, 2))
    ;create all hours array
    hh[j]=long(strmid(b_name, 8, 2))
    ;create all_months array
    minutes[j]=long(strmid(b_name, 10, 2))
endfor

```

```

;create array holding sorted, unique values of years
years= all_years[uniq(all_years)]

```

```

;(C)calculate nr of days in year and total expected images in the batch
;WARNING: the expected nr of images is calculated based on the goal of making
COMPLETE YEAR BATCHES.
;this program won't just fill the gaps from the first to the last image

```

```

    nr_images=0
    a= where (years ne 0, count)
    for k=0, count-1 do begin
        days_in_year=365

        if ((years[k] mod 4 eq 0) && (years[k] mod 100 ne 0)) then days_in_year=366
        if ((years[k] mod 4 eq 0) && (years[k] mod 100 eq 0) && (years[k] mod 400 eq 0)) then
days_in_year=366

        ;calculate nr of images expected, depending on image source
        nr_images=nr_images+days_in_year*24*hourly_images
    endfor

    print, 'nr_images=', nr_images

;(D) create input array, all values NaNs, seperate array for names
data_array = make_array(ns,nl,nr_images, /float, value= !values.f_nan)

;(E) fill in the input array with the existing images where they exist

;calculate the name components of the first expected image
expected_min=00
expected_hr=00
expected_dd=01
expected_mm=01
expected_yr=years[0]

;creat bname NaN string array to hold the names of the existing bands in the place where they're
written
bnames= make_array(nr_images, /string, value= string(0))

;position counter
w=0

;image loop
for y=0, nel-1 do begin

    jump: print, 'next position=', w

    if (w le nr_images-1) then begin

        print, expected_yr, expected_mm, expected_dd, expected_hr, expected_min

        ;calculate differences
        dmin= minutes[y]-expected_min
        dhrrs= hh[y]-expected_hr

```

```

ddays= dd[y]-expected_dd
dmon= mon[y]-expected_mm
dyr= all_years[y]-expected_yr

print, dyr,dmon,ddays, dhls, dmin

;calculate d=maximum days of expected month for subsequent calculations
case expected_mm of
  01:d=31
  02:d=28
  03:d=31
  04:d=30
  05:d=31
  06:d=30
  07:d=31
  08:d=31
  09:d=30
  10:d=31
  11:d=30
  12:d=31
endcase

days_of_exp_year=365
if ((expected_yr mod 4 eq 0) && (expected_yr mod 100 ne 0)) then days_of_exp_year=366
if ((expected_yr mod 4 eq 0) && (expected_yr mod 100 eq 0) && (expected_yr mod 400 eq 0))
then days_of_exp_year=366

if ((days_of_exp_year eq 366) && (expected_mm eq 02)) then d=29

;calculate subsequent expected image name components

if ((hourly_images eq 2) && (expected_min eq 00)) then expected_min=30 else begin
if ((hourly_images eq 2) && (expected_min ne 00)) then begin
expected_min= 00
if (expected_hr ne 23) then (expected_hr=expected_hr+1) else begin
expected_hr= 00
if (expected_dd ne d) then (expected_dd=expected_dd+1) else begin
expected_dd=1
if (expected_mm ne 12) then expected_mm=expected_mm+1 else begin
expected_mm=1
expected_yr= expected_yr+1
endelse
endelse
endif
endelse

```



```

    if (hourly_images eq 4) && (expected_min ne 45) then expected_min=expected_min+15 else
begin
    if ((hourly_images eq 4) && (expected_min eq 45)) then begin
        expected_min= 00
        if (expected_hr ne 23) then expected_hr=expected_hr+1 else begin
            expected_hr= 00
            if (expected_dd ne d) then expected_dd=expected_dd+1 else begin
                expected_dd=1
                if (expected_mm ne 12) then expected_mm=expected_mm+1 else begin
                    expected_mm=1
                    expected_yr= expected_yr+1
                endelse
            endelse
        endelse
    endif
endelse

```

```

print, expected_yr, expected_mm, expected_dd, expected_hr, expected_min

```

```

;if image y matches the expected then write it in position w of stack
;and go to check another image (exit position w for loop)
;else repeat loop (let w=w+1) calculating subsequent expected till you find the position w
;where image y components match the expected ones

```

```

if ((dyr eq 0) && (dmon eq 0) &&(ddays eq 0) && (dhrs eq 0) && (dmin eq 0)) eq 0 then begin
w=w+1
goto, jump
endif

```

```

if ((dyr eq 0) && (dmon eq 0) &&(ddays eq 0) && (dhrs eq 0) && (dmin eq 0)) eq 1 then begin
data_array[:,*,w]=read_tiff(file_names[y]) ;suddenly gives error: "array subscript for data_array
must have same size as source expression"
bnames[w]=b_names[y]
w=w+1
endif

```

```

endif

```

```

endfor ; back to y

```

```

print, 'loading done'

```

```

;(F) if mfg perform the calibration

if hourly_images eq 2 then begin

;first make array of only existing images
data=fltarr(ns, nl, nel)

;write the array of existing images
f=0
for e=0, nr_images-1 do begin
  if f le (nel-1) then begin
    if (bnames[e] ne 0) then begin
      data[*,*,f]=data_array[*,*,e]
      f=f+1
    endif
  endif
endif
endfor; back to e, i.e. change stack position

;calibrate the array of existing images
; get meteosat 5 calibration data
org_calibration = hw_shaky_caldata_mfg()

; create empty array for calibration data
; to be resampled to selected dates
res_calibration = fltarr(2,nel)

imgdates = float(strmid(b_names,0,12))
caldates = org_calibration[0,*]

starting = where(caldates ge imgdates[0],starting_exist)
if starting_exist ge 0 && starting[0] ne 0 then starting = starting[0]-1 else starting = 0
ending = where(caldates le imgdates[nel-1],ending_exist)
if ending_exist ge 0 then ending = (reverse(ending))[0]+1 else ending = n_elements(caldates)-1

; loop through calibration dates
for i = starting, ending-1 do begin
  ; check where image data is greater or equal to a calibration date
  index = where(imgdates ge caldates[i],count)

  ; in case a match is found...
  if (count gt 0) then begin
    ;print, imgdates[index[0],index[-1]], caldates[i] this used not to be semicolonated
    ; ...add the calibration date to the resampled calibration array
    res_calibration[*,index] = rebin(org_calibration[[1,2],i],2,count)
  endif
endif

```

```

endif
endfor

print, 'resampling done'

; convert to radiance
data = data - transpose(rebin(reform(res_calibration[1,*]),nel,ns,nl),[1,2,0])
data = data * transpose(rebin(reform(res_calibration[0,*]),nel,ns,nl),[1,2,0])

print, 'calibration done'

; convert to brightness temperature
A = 6.7348
B = -1272.2
;BT = B / (alog(radiance) - A)

data = alog(data)
data = B / (data - A)

print, 'conversion done'

;return the calibrated images in their correct place in the stack

v=0 ;image counter

; z position in data_array loop
for z=0, nr_images-1 do begin
  if v le (nel-1) then begin
    if b_names[v] eq bnames[z] then begin
      data_array[*,*,z]=data[*,*,v]
      v=v+1 ;check next image of small stack
    endif
  endif
endif
endfor ;back to z, i.e. check next big stack position

endif

;write the output file, whether mfg or msg
;remember to change the names :-P
openw,unit, 'name', /get_lun
writeu, unit, data_array
free_lun, unit

```

```

map_info = 'None'
envi_setup_head, $
  fname = 'name', $
  descrip = descrip, $
  ns = tif_info.dimensions[0], $
  nl = tif_info.dimensions[1], $
  nb = nr_images, $
  bbl = ok, $
  data_type = 4, $
  interleave = 0, $
  bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
  /write

```

```

print, 'import complete'

```

```

end

```

- Program for Polynomial Fit mask (also used for Polynomial Fit iterative method)  
pro mask\_polyfit, ev

```

compile_opt idl2

```

```

; INPUT

```

```

envi_open_file,/no_realize,r_fid=fid
if fid eq -1 then return
; query input image

```

```

envi_file_query,fid,dims=dims,ns=ns,nl=nl,nb=nb,bnames=bnames,fname=fname,data_type=da
ta_type

```

```

pos = lindgen(nb)
ns = dims[2]-dims[1]+1
nl = dims[4]-dims[3]+1

```

```

; create input array
input = fltarr(nl,nb,ns)

```

```

; loop through image lines
for l= 0,nl-1 do begin
  ; get a line out of the image and add to input array
  input[l,*,*] = envi_get_slice(fid=fid,line=l,pos=pos,/bip)
endfor

```

```

; close input file
envi_file_mng, id=fid, /remove

```

```

; transpose input data to bsq [ns,nl,nb]
input = transpose(input,[2,0,1])

polyfitThreshold = ftarr(ns,nl,nb)

;PROCESSING

;this has to be adjusted to the band naming of each dataset
; for Bam column-removed subset:
;fnames=strmid(bnames, 16,14)
; for van:
fnames=string(bnames)

;rescale the time-axis to a [0.01, nr_of_bands] range. Needs the scale_vector.pro and fpufix.pro
x = scale_vector(findgen(nb), 0.01, nb)

for i = 0,ns-1 do begin
  for j = 0,nl-1 do begin

    y = input[i,j,*]
    ;y = reform(y, nb)

    mask_nan = where(y gt 0, count)
    if count gt 0 then p = poly_fit(x[mask_nan],y[mask_nan],4)
    y_fit = poly(x,p)

    ;;Giovanni threshold
    mask_low = where(y lt y_fit, count)
    if count gt 0 then c = median(y_fit[mask_low]-y[mask_low])
    polyfitThreshold[i,j,*] = y_fit-c

    ;;other possible threshold(s):
;   mask_low = where(y lt y_fit, count)
;   if count gt 0 then thr=mean(y_fit[mask_low]-y[mask_low]) + 2*stddev(y_fit[mask_low]-
y[mask_low])
;   polyfitThreshold[i,j,*] = y_fit-thr
;etc

    ;;possible to replace poly_fit with robust_polyfit,
    ;;also needs robust_polyfit.pro, rob_checkfit.pro, robust_sigma.pro and med.pro
  endwhile
endfor

mask_temp = where(input lt polyfitThreshold, count)
if count gt 0 then input[mask_temp] = !values.f_nan

```

```

; OUTPUT
;change the names
openw,unit, 'name.bsq', /get_lun
writeu, unit, input
free_lun, unit
descrip = 'poly-masked data'
map_info = 'poly-masked data'
envi_setup_head, fname = 'name.bsq', descrip = descrip, ns = ns, nl = nl, nb = nb, $
data_type = 4, interleave = 0, bnames = bnames, wavelength_units = 5, map_info = map_info, $
/write

```

```

print, 'done'

```

- Program for pixel normalization  
pro kernel\_calc, ev

```

compile_opt idl2

```

```

title = 'Kernel_calc'

```

```

;; INPUT

```

```

envi_open_file,/no_realize,r_fid=fid
if fid eq -1 then return

```

```

; query input image
envi_file_query,fid,dims=dims,ns=ns,nl=nl,nb=nb,$
  bnames=bnames,data_type = data_type, fname=fname
pos = lindgen(nb)
ns = dims[2]-dims[1]+1
nl = dims[4]-dims[3]+1

```

```

;ns=8

```

```

;nl=6

```

```

;nb=3

```

```

; create input array
maskedfile = fltarr(nl,nb,ns)

```

```

; loop through image lines
for l= 0,nl-1 do begin
  ; get a line out of the image
  ; and add to input array
  maskedfile[l,*,*] = envi_get_slice(fid=fid,line=l,pos=pos,/bip)
endfor

```

```

print, 'loaded data'

; close input file
envi_file_mng, id=fid, /remove

; transpose input data to bsq [ns,nl,nb]
maskedfile = transpose(maskedfile,[2,0,1])

; create input array
testarray = fltarr(ns,nl)

; loop to load images, and fill data arrays
for b = 0, nb-1 do begin

    testarray[*,*] = (maskedfile[*,*],b)

;testing with a model array
;create array

;testarray= fltarr(8,6)
;testarray[*]= [1,0,1,!values.f_nan, !values.f_nan, 1,2,1]
;testarray[*]= [0,2,1,0,0,!values.f_nan, 1,2]
;testarray[*]= [!values.f_nan, !values.f_nan, !values.f_nan, 0, !values.f_nan, !values.f_nan,
!values.f_nan, !values.f_nan]
;testarray[*]= [2,2,1,1, !values.f_nan, 1, 1, !values.f_nan]
;testarray[*]= [0,1,0,2,!values.f_nan, 1, 2, !values.f_nan]
;testarray[*]= [1,2,1,0,!values.f_nan,!values.f_nan,!values.f_nan,!values.f_nan]

;set kernelsize
ks= 12

; ; create kernel for normalization
; kernel = replicate(1b,kernelsize*2+1,kernelsize*2+1)
; kernel[1:kernelsize*2-1,1:kernelsize*2-1] = 0
;
;
; max_kernel_pixel_nr=where(kernel ne 0, count)
; print, count

;calculate expected max nr of pixels in the kernel
;kernel_pixels=4*(2*kernelsize+1)-4

kernelvalue=0
NaN_count=0
kp_count=0 ;number of existing kernel pixels
;cs= 6 ;centerpixel sample

```



```

;cl= 4 ;centerpixel line

;ns=8
;nl=6

;create output file to store normalized pixel values
output=fltarr(ns, nl)

for cl=0, nl-1 do begin
for cs=0, ns-1 do begin

;general kernel scheme
if ((cs ge ks) && (cs le (ns-1-ks)) && (cl ge ks) && (cl le (nl-1-ks))) then begin

;left side
for j= cl-ks, cl+ks do begin

    if testarray[cs-ks, j] eq testarray[cs-ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs-ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs-ks, j]=0
        kernelvalue=kernelvalue+testarray[cs-ks, j]
        testarray[cs-ks, j]=!values.f_nan
    endelse

;right side

    if testarray[cs+ks, j] eq testarray[cs+ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs+ks, j]=0
        kernelvalue=kernelvalue+testarray[cs+ks, j]
        testarray[cs+ks, j]=!values.f_nan
    endelse

endfor

;upper part
for i= cs-(ks-1), cs+(ks-1) do begin

    if testarray[i, cl-ks] eq testarray[i, cl-ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl-ks]
    endif else begin

```

```

        NaN_count=NaN_count+1
        testarray[i, cl-ks]=0
        kernelvalue=kernelvalue+testarray[i, cl-ks]
        testarray[i, cl-ks]=!values.f_nan
    endelse

;lower part

    if testarray[i, cl+ks] eq testarray[i, cl+ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl+ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl+ks]=0
        kernelvalue=kernelvalue+testarray[i, cl+ks]
        testarray[i, cl+ks]=!values.f_nan
    endelse
endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

;left middle part of margin
if ((cs lt ks) && (cl ge ks) && (cl le (nl-1-ks))) then begin

;right side
for j= cl-ks, cl+ks do begin

    if testarray[cs+ks, j] eq testarray[cs+ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs+ks, j]=0
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endelse
endfor

```

```

        testarray[cs+ks, j]=!values.f_nan
    endelse

endfor

;upper part
for i= 0, cs+(ks-1) do begin

    if testarray[i, cl-ks] eq testarray[i, cl-ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl-ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl-ks]=0
        kernelvalue=kernelvalue+testarray[i, cl-ks]
        testarray[i, cl-ks]=!values.f_nan
    endelse

;lower part

    if testarray[i, cl+ks] eq testarray[i, cl+ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl+ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl+ks]=0
        kernelvalue=kernelvalue+testarray[i, cl+ks]
        testarray[i, cl+ks]=!values.f_nan
    endelse
endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

; right middle margin
if ((cs gt (ns-1-ks)) && (cl ge ks) && (cl le (nl-1-ks))) then begin

```

```

;left side
for j= cl-ks, cl+ks do begin

    if testarray[cs-ks, j] eq testarray[cs-ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs-ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs-ks, j]=0
        kernelvalue=kernelvalue+testarray[cs-ks, j]
        testarray[cs-ks, j]=!values.f_nan
    endelse

endfor

;upper part
for i= cs-(ks-1), ns-1 do begin

    if testarray[i, cl-ks] eq testarray[i, cl-ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl-ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl-ks]=0
        kernelvalue=kernelvalue+testarray[i, cl-ks]
        testarray[i, cl-ks]=!values.f_nan
    endelse

endfor

;lower part

    if testarray[i, cl+ks] eq testarray[i, cl+ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl+ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl+ks]=0
        kernelvalue=kernelvalue+testarray[i, cl+ks]
        testarray[i, cl+ks]=!values.f_nan
    endelse

endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count

```

```

;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

;upper middle margin
if ((cs ge ks) && (cs le (ns-1-ks)) && (cl lt ks)) then begin

;left side
for j= 0, cl+ks do begin

    if testarray[cs-ks, j] eq testarray[cs-ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs-ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs-ks, j]=0
        kernelvalue=kernelvalue+testarray[cs-ks, j]
        testarray[cs-ks, j]=!values.f_nan
    endelse

;right side

    if testarray[cs+ks, j] eq testarray[cs+ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs+ks, j]=0
        kernelvalue=kernelvalue+testarray[cs+ks, j]
        testarray[cs+ks, j]=!values.f_nan
    endelse

endifor

;lower part
for i= cs-(ks-1), cs+(ks-1) do begin

    if testarray[i, cl+ks] eq testarray[i, cl+ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl+ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl+ks]=0
    endelse
endfor

```

```

        kernelvalue=kernelvalue+testarray[i, cl+ks]
        testarray[i, cl+ks]=!values.f_nan
    endelse
endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

;lower middle margin
if ((cs ge ks) && (cs le (ns-1-ks)) && (cl gt (nl-1-ks))) then begin

;left side
for j= cl-(ks-1), nl-1 do begin

    if testarray[cs-ks, j] eq testarray[cs-ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs-ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs-ks, j]=0
        kernelvalue=kernelvalue+testarray[cs-ks, j]
        testarray[cs-ks, j]=!values.f_nan
    endelse

;right side

    if testarray[cs+ks, j] eq testarray[cs+ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs+ks, j]=0
        kernelvalue=kernelvalue+testarray[cs+ks, j]
        testarray[cs+ks, j]=!values.f_nan
    endelse

endif

endfor

```

```

;upper part
for i= cs-ks, cs+ks do begin

    if testarray[i, cl-ks] eq testarray[i, cl-ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl-ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl-ks]=0
        kernelvalue=kernelvalue+testarray[i, cl-ks]
        testarray[i, cl-ks]=!values.f_nan
    endelse

endif

endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

; up left corner
if ((cs lt ks) && (cl lt ks)) then begin

;right side
for j= 0, cl+ks do begin

    if testarray[cs+ks, j] eq testarray[cs+ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs+ks, j]=0
        kernelvalue=kernelvalue+testarray[cs+ks, j]
        testarray[cs+ks, j]=!values.f_nan
    endelse

endif

endfor

```



```

;lower part
for i= 0, cs+(ks-1) do begin

    if testarray[i, cl+ks] eq testarray[i, cl+ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl+ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl+ks]=0
        kernelvalue=kernelvalue+testarray[i, cl+ks]
        testarray[i, cl+ks]=!values.f_nan
    endelse
endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

;low left corner
if ((cs lt ks) && (cl gt (nl-1-ks))) then begin

;right side
for j= cl-ks, nl-1 do begin

    if testarray[cs+ks, j] eq testarray[cs+ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs+ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs+ks, j]=0
        kernelvalue=kernelvalue+testarray[cs+ks, j]
        testarray[cs+ks, j]=!values.f_nan
    endelse

endfor

;upper part

```

```

for i= 0, cs+(ks-1) do begin

    if testarray[i, cl-ks] eq testarray[i, cl-ks] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl-ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl-ks]=0
        kernelvalue=kernelvalue+testarray[i, cl-ks]
        testarray[i, cl-ks]=!values.f_nan
    endelse

endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

;upper right corner
if ((cs gt (ns-1-ks)) && (cl lt ks)) then begin

;left side
for j= 0, cl+ks do begin

    if testarray[cs-ks, j] eq testarray[cs-ks, j] then begin
        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[cs-ks, j]
    endif else begin
        NaN_count=NaN_count+1
        testarray[cs-ks, j]=0
        kernelvalue=kernelvalue+testarray[cs-ks, j]
        testarray[cs-ks, j]=!values.f_nan
    endelse

endfor

;lower part
for i= cs-(ks-1),ns-1 do begin

```

```

    if testarray[i, cl+ks] eq testarray[i, cl+ks] then begin
      kp_count=kp_count+1
      kernelvalue=kernelvalue+testarray[i, cl+ks]
    endif else begin
      NaN_count=NaN_count+1
      testarray[i, cl+ks]=0
      kernelvalue=kernelvalue+testarray[i, cl+ks]
      testarray[i, cl+ks]=!values.f_nan
    endelse

  endfor

  if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
  if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

  ;print, 'kernelvalue=', kernelvalue
  ;print, 'nan=', NaN_count
  ;print, 'kernel pixels=', kp_count
  ;print, 'kernel=', kernel

  kernelvalue=0
  NaN_count=0
  kp_count=0

endif

;low right corner
if ((cs gt (ns-1-ks)) && (cl gt (nl-1-ks))) then begin

;left side
for j= cl-ks, nl-1 do begin

  if testarray[cs-ks, j] eq testarray[cs-ks, j] then begin
    kp_count=kp_count+1
    kernelvalue=kernelvalue+testarray[cs-ks, j]
  endif else begin
    NaN_count=NaN_count+1
    testarray[cs-ks, j]=0
    kernelvalue=kernelvalue+testarray[cs-ks, j]
    testarray[cs-ks, j]=!values.f_nan
  endelse

endfor

;upper part
for i= cs-(ks-1), ns-1 do begin

  if testarray[i, cl-ks] eq testarray[i, cl-ks] then begin

```

```

        kp_count=kp_count+1
        kernelvalue=kernelvalue+testarray[i, cl-ks]
    endif else begin
        NaN_count=NaN_count+1
        testarray[i, cl-ks]=0
        kernelvalue=kernelvalue+testarray[i, cl-ks]
        testarray[i, cl-ks]=!values.f_nan
    endelse

endfor

if kp_count eq 0 then kernel= !values.f_nan else kernel= kernelvalue/kp_count
if kernel eq 0 then output[cs,cl]= !values.f_nan else output[cs,cl]= testarray[cs,cl]/kernel

;print, 'kernelvalue=', kernelvalue
;print, 'nan=', NaN_count
;print, 'kernel pixels=', kp_count
;print, 'kernel=', kernel

kernelvalue=0
NaN_count=0
kp_count=0

endif

endfor
endfor

maskedfile[*,*,b]=output[*,*]

endfor

;print, maskedfile[*,*,0]
;print, maskedfile[*,*,1]
;print, maskedfile[*,*,2]

;; OUTPUT

; output to regular file
openw,unit, 'name.bsq' , /get_lun
writeu, unit, maskedfile
free_lun, unit

descrip = 'test stack normalized data'
map_info = 'None'
envi_setup_head, $
    fname = 'name.bsq', $

```

```

descrip = descrip, $
ns = ns, $
nl = nl, $
nb = nb, $
data_type = 4, $
interleave = 0, $
bnames = bnames, $
wavelength_units = 5, $
map_info = map_info, $
/write

```

```

print, 'normalization done'

```

```

end

```

- Program for anomaly flagging and anomaly count

```

;+

```

```

; hw_shaky_filt, v20110328.01

```

```

;-

```

```

pro hw_shaky_filt, ev

```

```

compile_opt idl2

```

```

catch, theerror

```

```

if theerror ne 0 then begin

```

```

    catch, /cancel

```

```

    message = [!error_state.msg]

```

```

    void = dialog_message(message, /center, /error)

```

```

    return

```

```

endif

```

```

title = 'Shaky'

```

```

;; INPUT

```

```

envi_open_file,/no_realize,r_fid=fid

```

```

if fid eq -1 then return

```

```

; query input image

```

```

envi_file_query,fid,dims=dims,ns=ns,nl=nl,nb=nb,$

```

```

    bnames=bnames,data_type=data_type,fname=fname

```

```

pos = lindgen(nb)

```

```

ns = dims[2]-dims[1]+1

```

```

nl = dims[4]-dims[3]+1

```

```

; create input array
input = fltarr(nl,nb,ns)

; loop through image lines
for l= 0,nl-1 do begin

    ; get a line out of the image
    ; and add to input array
    input[l,*,*] = envi_get_slice(fid=fid,line=l,pos=pos,/bip)

endfor

print, 'loaded data'

; close input file
envi_file_mng, id=fid, /remove

; transpose input data to bsq [ns,nl,nb]
input = transpose(input,[2,0,1])

;; GUI

; create default output names
fname = file_basename(fname)
ext = strpos(fname,'./reverse_offset/reverse_search)
fname = strmid(fname,0,ext)
if fname eq " then fname='shaky'
out_file = fname+'_filt'

; dirty dirty... should be done better
; case fix( (double(bnames[1]) - double(bnames[0])) ) of
;   1500 : hourly = 4
;   3000 : hourly = 2
;   10000 : hourly = 1
;   20000 : hourly = 0.5
;   else : hourly = 1
; endcase

; number of images recorded every hour
hourly=2

; start gui
base0 = widget_auto_base(title=title)

base1 = widget_base(base0,/column,/frame,/grid_layout)
base2 = widget_base(base1,/row)
void = widget_slider(base2,title='kernel size (days)', $

```

```

max=14,min=1,value=5,uvalue='tempksize',/auto)

base1 = widget_base(base0,/column,/frame,/grid_layout)
base2 = widget_base(base1,/row)
void = widget_slider(base2,title='Kernel radius (px)', $
    max=10,min=1,value=2,uvalue='spatksize',/auto)

base1 = widget_base(base0,/column,/frame,/grid_layout)
base2 = widget_base(base1,/column)
void = widget_outf(base2, prompt = 'Output basename', $
    default=out_file,uvalue = 'out_name', /auto)
set = auto_wid_mng(base0)

if set.accept eq 0 then return

```

```
;; TEMPORAL FILTERING
```

```

; define temporal kernel
kernel = intarr(set.tempksize*24*hourly) + 1

```

```

;reform input image to 2D
input = reform(input,ns*nl,nb,/overwrite)

```

```

; flag valid (non-margin) and invalid (margin) pixels
totfin = total(finite(input),2,/double)
valid = where(totfin gt 0,nvalid)
invalid = where(totfin eq 0,ninvalid)
totfin = 0b

```

```

;HERE added output 5 finite and removed spatial filtering
; define output arrays
output1 = bytarr(ns*nl,nb) ; flagged pixels
output2 = intarr(ns*nl,nb) ; time convolution
output5 = intarr(ns*nl,nb) ; finite counter in time

```

```

; loop through all pixels that do not belong to margin
for i = 0L, nvalid-1 do begin

```

```

    ;extract tixel (spectrum through time)
    tixel = reform(input[valid[i],*])

```

```

    ; flag values greater than mean + 2 stdev
    mom = moment(tixel,maxmoment=2,/nan)
    tixel = tixel gt (mom[0]+2*sqrt(mom[1]))
    ;tixel = tixel gt (mean(tixel,/nan)+2*stddev(tixel,/nan))

```



```

; save flags to output array
output1[valid[i],*] = byte(tixel)

; save temporal kernel convolution to output array
output2[valid[i],*] = convol(tixel,kernel,invalid=0,missing=0)

;HERE IS ADDITIONAL FINITE DEFINITION
; save finite temporal kernel convolution to output array
output5[valid[i],*] = convol(finite(tixel),kernel,invalid=0,missing=0)

endfor

; loose variable to save memory
input = 0b

;reform output image back to 3D
output1 = reform(output1,ns,nl,nb,/overwrite)
output2 = reform(output2,ns,nl,nb,/overwrite)
output5 = reform(output5,ns,nl,nb,/overwrite)

print, 'temporal filtering done'

;HERE is the additional product of max flagged counts
;all spatial filtering is discarded
tempmax=max(output2, dimension=3)

;; OUTPUT

; output to image
openw,unit, set.out_name+'_flag.bsq', /get_lun
writeu, unit, output1
free_lun, unit

descrip = 'Shaky - anomaly flagged data'
map_info = 'None'
envi_setup_head, $
  fname = set.out_name+'_flag.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = nb, $
  data_type = 1, $
  interleave = 0, $
  bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
/write

```

```
; output to image
openw,unit, set.out_name+'_temp.bsq', /get_lun
writeu, unit, output2
free_lun, unit
```

```
descrip = 'Shaky - temporally filtered data'
map_info = 'None'
envi_setup_head, $
  fname = set.out_name+'_temp.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = nb, $
  data_type = 2, $
  interleave = 0, $
  bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
  /write
```

```
; output to regular image
openw,unit, set.out_name+'_finitetemp.bsq', /get_lun
writeu, unit, output5
free_lun, unit
```

```
descrip = 'Shaky - temporal finite pixels'
map_info = 'None'
envi_setup_head, $
  fname = set.out_name+'_finitetemp.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = nb, $
  data_type = 2, $
  interleave = 0, $
  bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
  /write
```

```
openw,unit, set.out_name+'_tempmax.bsq', /get_lun
writeu, unit, tempmax
free_lun, unit
```

```
descrip = 'Shaky - max temp data'
```

```

map_info = 'None'
envi_setup_head, $
  fname = set.out_name+'_tempmax.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = 1, $
  data_type = 2, $
  interleave = 0, $
; bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
  /write

```

```

print, 'filtering done'

```

- Program for correction for missing values  
pro correction, ev

```

compile_opt idl2

```

```

catch, theerror
if theerror ne 0 then begin
  catch, /cancel
  message = [!error_state.msg]
  void = dialog_message(message, /center, /error)
  return
endif

```

```

title = 'Shaky'

```

```

;; INPUT

```

```

envi_open_file,/no_realize,r_fid=fid
if fid eq -1 then return

```

```

; query input image
envi_file_query,fid,dims=dims,ns=ns,nl=nl,nb=nb,$
  bnames=bnames,data_type=data_type,fname=fname
pos = lindgen(nb)
ns = dims[2]-dims[1]+1
nl = dims[4]-dims[3]+1

```

```

; create input array
input = fltarr(nl,nb,ns)

```

```

; loop through image lines

```

```

for l= 0,nl-1 do begin

    ; get a line out of the image
    ; and add to input array
    input[l,*,*] = envi_get_slice(fid=fid,line=l,pos=pos,/bip)

endfor

print, 'loaded data'

; close input file
envi_file_mng, id=fid, /remove

; transpose input data to bsq [ns,nl,nb]
input = transpose(input,[2,0,1])

;nr of observations in temporal window
obs=336

;create 2D image with the mean+2sd threshold
tixel=fltarr(1,nb)
t=fltarr(ns,nl)

for j=0,nl-1 do begin
for k=0,ns-1 do begin
tixel=input[k,j,*]
mom = moment(tixel,maxmoment=2,/nan)
t[k,j]=mom[0]+2*sqrt(mom[1])
print, t[k,j]
endfor
endfor

;flagging
for i=0, nb-1 do begin
for j=0,nl-1 do begin
for k=0,ns-1 do begin
if input[k,j,i] eq input [k,j,i] then begin
; print, 'inp=',input[k,j,i]
if input[k,j,i] ge t[k,j] then input[k,j,i]=1 else input[k,j,i]=0
; print, 'input=', input[k,j,i]
endif else begin
input[k,j,i]=!values.f_nan
endelse
endfor
endfor
endfor
endfor

```

```

print, 'done flagging'

;counting flags and finite
peaks=fltarr(ns,nl,nb)
finites=fltarr(ns,nl,nb)

for i=0,335 do begin
input[*,*,i]=0
endfor
for i=336, nb-1 do begin
for j=0,nl-1 do begin
for k=0,ns-1 do begin
array=(input[k,j,i-336:i])
peaks[k,j,i]=total(array,/NAN)
; print, 'peaks=',peaks[k,j,i]
fin=where(finite(array), count)
finites[k,j,i]=count
; print, 'fins=', finites[k,j,i]
endfor
endfor
endfor

print, 'done counting'

; correcting for data availability
corrected=fltarr(ns,nl,nb)
for i=0, nb-1 do begin
for j=0,nl-1 do begin
for k=0,ns-1 do begin
corrected[k,j,i]=peaks[k,j,i]/(finites[k,j,i]/337)
endfor
endfor
endfor

;calculating max
tempmax=max(peaks, dimension=3)

;; OUTPUT'

; output to image
openw,unit, 'name.bsq', /get_lun
writeu, unit, peaks
free_lun, unit

descrip = 'Shaky - temporally filtered data'
map_info = 'None'
envi_setup_head, $
    fname = 'name.bsq', $

```

```

descrip = descrip, $
ns = ns, $
nl = nl, $
nb = nb, $
data_type = 4, $
interleave = 0, $
bnames = bnames, $
wavelength_units = 5, $
map_info = map_info, $
/write

```

```

; output to regular image
openw,unit, 'name.bsq', /get_lun
writeu, unit, finites
free_lun, unit

```

```

descrip = 'Shaky - temporal finite pixels'
map_info = 'None'
envi_setup_head, $
  fname = 'name.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = nb, $
  data_type = 4, $
  interleave = 0, $
  bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
/write

```

```

; output to regular image
openw,unit, 'name.bsq', /get_lun
writeu, unit, corrected
free_lun, unit

```

```

descrip = 'Shaky - peaks corrected for data availability'
map_info = 'None'
envi_setup_head, $
  fname = 'name.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = nb, $
  data_type = 4, $
  interleave = 0, $
  bnames = bnames, $

```

```

wavelength_units = 5, $
map_info = map_info, $
/write

openw,unit, 'name.bsq', /get_lun
writeu, unit, tempmax
free_lun, unit

descrip = 'Shaky - max temp data'
map_info = 'None'
envi_setup_head, $
  fname = 'name.bsq', $
  descrip = descrip, $
  ns = ns, $
  nl = nl, $
  nb = 1, $
  data_type = 4, $
  interleave = 0, $
; bnames = bnames, $
  wavelength_units = 5, $
  map_info = map_info, $
  /write

print, 'filtering done'

end

```



