

INTEGRATION OF EMBEDDED AND REMOTE SENSED TEMPERATURE FOR
DAILY TEMPERATURE MAPPING

By

FARBOD FARZAN

A thesis submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Master of Science

Graduate Program in Civil and Environmental Engineering

written under the direction of

David J. Hill

and approved by

New Brunswick, New Jersey

October 2011

© 2011

FARBOD FARZAN

ALL RIGHTS RESERVED

ABSTRACT OF THE THESIS

Integration of Embedded and Remote Sensed Temperature for Daily Temperature Mapping

by FARBOD FARZAN

Thesis Director:

David J. Hill

Temperature variability is an important driver of many important global and regional processes, which has inspired researchers to understand and predict the spatial variability of surface air temperature. This importance has increased demand for quality, high resolution gridded climatological datasets that deliver detailed information on the variability of temperature at regional scales. Several interpolation and extrapolation techniques have been introduced that use point data sources (land-based data from weather stations). However, the scarcity of weather stations with long-term records and good spatial coverage and the impacts of a non-stationary climate limits these traditional methods. Through the analysis of existing ground-based temperature sensors we have shown that there are inadequate ground-based measurements to estimate the spatial variability of daily min/max temperature. Furthermore, we have shown that existing interpolation methods are insufficiently accurate to estimate the local temperature at ungauged locations because they cannot capture anthropogenic (e.g. urban heat island) or microclimatological (e.g. cold air pooling) effects. This result implies that, in general,

ground-based temperature measurements are too sparse to capture the spatial variability of temperature. Together with satellite observations, gridded meteorological variables can provide important information of the complex interactions of these features in order to accurately map temperature across broad regions. Satellite remote-sensing is another way for acquisition of land surface temperature (LST) data. However, due to technical constraints, satellite thermal sensors are incapable to supply both spatially and temporally dense LST image data. The reason for this is that the spatial and temporal resolutions of a satellite thermal sensor are anti-correlated, meaning that a high spatial resolution is related with low temporal resolution and vice versa. The trade-off between spatial and temporal resolution of satellite data, encouraged us to apply the Moderate Resolution Imaging Spectroradiometer (MODIS) as a source of remote-sensed land surface temperature data to capture many of the rapid biological and meteorological changes that MODIS (Spatial Resolution [bands 20-23]: 1km, 5km) observes in every 1 to 2 days. This work integrates both remote sensed and ground-based observations to account for anthropogenic and microclimatological impacts on the surface air temperature. Our method combines land-based embedded sensor data with remote-sensed data to construct continuous maps of daily min/max temperature over broad regions, and is able to capture the underlying spatial variability of temperature better than other traditional spatial methods.

Acknowledgment

The author wishes to thank several people. I would like to thank my sister, Farnaz, for her love, support and patience during the years it has taken me to graduate. I would like to thank my parents for their unending love and support. I would also like to thank Dr. Hill for all his generous helps and guidance which have inspired me during the last two years. It was his direction which helped me to accomplish this project. Last but not least, I would like to thank all my committee members who kindly advised me, and put their precious time toward my thesis.

Table of Contents

1	Introduction to Spatial Interpolation.....	xiv
1.1	Introduction	xiv
1.2	Scope of Thesis	xx
2	Study of Temperature Data over the State of New Jersey.....	xxii
2.1	Introduction	xxii
2.2	Seasonality Removal	xxv
2.2.1	Statistical analysis for existence of seasonality	xxvi
2.3	Semivariogram Analysis	xxix
2.4	Temperature modeling challenges	xxxii
2.5	Conclusion.....	xxxvi
3	Inverse Distance Weighting vs. Truncated Gaussian Filter	xxxvii
3.1	Introduction	xxxvii
3.1.1	Inverse Distance Weighting (IDW)	xxxviii
3.1.2	Truncated Gaussian Filter (TGF).....	xxxviii
3.2	Interpolation Implementation.....	xli
3.3	Test validation procedure: Mean Square Error (MSE)	xlii
3.4	Parameterizing of Models Over Peaks Data	xliv
3.4.1	Truncated Gaussian Filter Parameter Justification	xliv
3.4.2	Inverse Distance Weighting Parameter Justification	xliv
3.5	Test of Interpolation Performance.....	xlvi
3.6	Defining a More Realistic Surface	l
3.7	Test of Interpolation over regions with “Sparse & Clusters” of sensors	lii
3.8	Defining the number of nearest neighbors in IDW	liii
3.9	Conclusion.....	lvi
4	Remote Sensing and its Functionality in Retrieving Land Surface Temperature	lvii
4.1	Introduction	lvii
4.2	Earth Observing System (EOS)	lix
4.3	Moderate Resolution Imaging Spectrophotometer (MODIS).....	lx

4.3.1	MODIS technical Specifications.....	lxii
4.3.2	File Format of land surface temperature (LST) Products	lxiv
4.3.3	MOD 11 - Land surface temperature (LST) and Emissivity	lxv
4.4	Image Processing.....	lxvii
4.4.1	Converting HDF formatted files to GeoTIFF images.....	lxvii
4.4.2	MODIS outliers.....	lxviii
4.5	Ground-Based Temperature (TG) Vs. MODIS-Based Temperature (TM).....	lxxi
4.6	Mathematical Integration of Ground-based & Remote-sensed Temperature Data.....	lxxvii
4.7	Test of spatial dependency	lxxviii
4.8	Error distribution over the state of New Jersey.....	lxxix
4.9	Mathematical function of temperature interpolator model	lxxx
4.10	Leave one out cross validation Test	lxxxii
4.11	Model Demonstration.....	lxxxiv
4.12	Conclusion and results	lxxxix
5	Conclusion and Discussion.....	xc
5.1	Future Work	xcii
I	Appendix	xciii
II	References	cxviii

List of tables

Table 1 Average maximum, minimum, and mean temperature over the state of New Jersey (1/1/1995 – 1/1/2008)	xxiii
Table 2 Availability of data in New Jersey meteorological stations	xxiv
Table 3 Result summary of parameter justifications of IDW and TGF.....	xliv
Table 4 MSE values IDW and TGF.....	1
Table 5 MSE values for inclined surface and Gaussian surface.....	liii
Table 6 MSE values from IDW and M-IDW	lv
Table 7 MODIS technical Specifications	lxii
Table 8 MODIS LST data products	lxiv
Table 9 Results of spatial dependency test	lxxix

List of illustrations

Figure 1 Distribution of sensors over the state of New Jersey	xxiii
Figure 2 Ten-year average of maximum daily temperature at each cooperative network sensor location	xxv
Figure 3 Autocorrelation plot for ten-year of maximum temperature	xxvii
Figure 4 De-trended ten-year average maximum temperature at each cooperative network sensor location	xxviii
Figure 5 Partial autocorrelation plot for ten-year maximum temperature	xxix
Figure 6 Sample semivariogram of data without significant spatial trends.....	xxxi
Figure 7 The semivariogram plot for sensors	xxxii
Figure 8 De-trended maximum temperature square versus elevation	xxxiii
Figure 9 De-trended minimum temperature versus elevation.....	xxxiii
Figure 10 De-trended maximum temperature versus log of population density	xxxiv
Figure 11 De-trended minimum temperature versus log of population density	xxxiv
Figure 12 De-trended maximum temperature versus northing	xxxv
Figure 13 De-trended minimum temperature versus northing.....	xxxv
Figure 14 Conventional peaks function	xlii
Figure 15 MSE distribution TGF (10 sensors included).....	xliii
Figure 16 MSE distribution IDW (10 sensors included)	xliii
Figure 17 MSE value IDW Number of sensors: 10.....	xlvi
Figure 18 MSE vale TGF Number of sensors: 10	xlvi
Figure 19 MSE value IDW, Number of sensors: 20.....	xlvii
Figure 20 MSE value TGF, Number of sensors: 20	xlvii

Figure 21 MSE value IDW, Number of sensors: 40.....	xlvi
Figure 22 MSE value TGF, Number of sensors: 40	xlvi
Figure 23 MSE value IDW, Number of sensors: 640.....	xlvi
Figure 24 MSE value TGF, Number of sensors: 640	xlvi
Figure 25 Sensor value Vs. interpolated value from IDW.....	xlvii
Figure 26 Sensor value Vs. interpolated value from TGF	xlvii
Figure 27 Surface with Gaussian distributions	li
Figure 28 Inclined surface	li
Figure 29 Sparse and cluster distribution of sensors	lii
Figure 30 IDW and modified-IDW flowchart	liv
Figure 31 Center of cluster and contributed sensors.....	lv
Figure 32 Gridded tile map projection format	lxvi
Figure 33 New Jersey location with respect to tile projection.....	lxvi
Figure 34 Terra h12v04, 08/05/2006	lxviii
Figure 35 Terra h12v05, 08/05/2006	lxviii
Figure 36 EWdata density plot	lxix
Figure 37 NSdata density plot.....	lxix
Figure 38 EWdata probability plot	lxx
Figure 39 NSdata probability plot.....	lxx
Figure 40 Atlantic city International Airport TG versus TM	lxxi
Figure 41 Beleplain Stn Forest TG versus TM.....	lxxi
Figure 42 Canoe Brook TG versus TM	lxxii
Figure 43 Cape May 2 NW TG versus TM	lxxii

Figure 44 Cranford TG versus TM	lxxii
Figure 45 Charlottesville TG versus TM.....	lxxii
Figure 46 Estell Manor TG versus TM.....	lxxiii
Figure 47 Flemington 5 NNW Tg versus TM	lxxiii
Figure 48 Free Hold-Marlboro TG versus TM.....	lxxiii
Figure 49 Hammonton 1 NE TG versus TM	lxxiii
Figure 50 Harrison TG versus TM.....	lxxiv
Figure 51 Hightstown 2 W TG versus TM.....	lxxiv
Figure 52 Indian Mills 2 W TG versus TM	lxxiv
Figure 53 Millville Municipal Airport TG versus TM	lxxiv
Figure 54 Moorestown TG versus TM	lxxv
Figure 55 New Brunswick 3 SE TG versus TM.....	lxxv
Figure 56 Newark International Airport TG versus TM.....	lxxv
Figure 57 Plainfield TG versus TM.....	lxxv
Figure 58 Sussex 2 NW TG versus TM.....	lxxvi
Figure 59 Seabrook Farms TG versus TM	lxxvi
Figure 60 Tuckerton TG versus TM.....	lxxvi
Figure 61 Toms River TG versus TM.....	lxxvii
Figure 62 Wertsville 4 NE TG versus TM.....	lxxvi
Figure 63 Residual distribution plot (Left: April, Right: May)	lxxx
Figure 64 Residual distribution plot (Left: June, Right: July)	lxxx
Figure 65 Residual distribution plot (Left: August, Right: September)	lxxxi
Figure 66 LOOCV plot, Terra Vs. TGmax.....	lxxxiii

Figure 67 LOOCV plot Aqua Vs. TGmax	lxxxiii
Figure 68 Raw MODIS data 04/02/2006	lxxxv
Figure 69 MODIS enhanced NN interpolated gauge data 04/02/2006.....	lxxxv
Figure 70 NN interpolated gauge data 04/02/2006.....	lxxxvi
Figure 71 IDW interpolated gauge data 04/02/2006.....	lxxxvi
Figure 72 Raw MODIS data 06/18/2006	lxxxvii
Figure 73 MODIS enhanced NN interpolated gauge data 06/18/2006.....	lxxxvii
Figure 74 NN interpolated gauge data 06/18/2006.....	lxxxviii
Figure 75 IDW interpolated gauge data 06/18/2006.....	lxxxviii

Chapter1

Introduction to Spatial Interpolation

1.1 Introduction

Spatial interpolation of climatic information has a significant role in local policy decisions and natural resource management. Numerous applications such as validation of climate models (Widmann and Bretherton 2000 [1]), monitoring or detecting and assessing potential impacts of regional climate change (Lobell et al. 2006 [2]), risk management (Kaplan and New 2006 [3]), and impact of human activities on regional environments and ecosystem services, are important factors in such a task. In the study of climate change, temperature is one of the most important variables, along with precipitation, and greenhouse gases like carbon dioxide. Generally there is considerable motivation to understand and predict the variability of regional surface air temperature. For instance, summer temperature, especially daily extremes, dictate electrical energy usage, affect water demand, and impact agriculture and ecosystems. Temperature variability is an important driver of many ecological processes, it has been understood for some time that variability in temperature and soil moisture is a major determinant of plant distribution (Whittaker, 1967 [4]; Stephenson, 1990 [5]). The numerous ecological

processes (photosynthesis, evapotranspiration, respiration, decomposition, etc.) are closely related to meteorological conditions.

Environmental forces such as precipitation and temperature have significant impact on the performance of infrastructure. In particular, the long-term performance of bridges is significantly impacted by the cyclical freezing and thawing of moisture. Freeze/thaw cycling degrades concrete and asphalt surfaces reducing the strength of bridge superstructures, exposes reinforcing steel to accelerated corrosion, and damages bridge deck surfaces. Thus, quantifying the freeze/thaw cycling for each bridge is a necessary step in evaluating and forecasting the performance of a bridge inventory. Such information could be derived from daily min/max temperatures measured at each bridge location with developing spatial models of the surface air temperature that can be used to estimate the temperature over broad areas. The goal of this particular case will be “virtual temperature sensors” that can be deployed at any location (i.e. at a bridge location) within the model domain. Virtual sensors have been explored in a number of scientific domains, and have been shown to be powerful tools for providing indirect measurements of phenomena of interest at ungauged locations (Cecil & Kozłowska 2009 [6]; Havlik et al. 2009 [7]; Douglas et al. 2008 [8]; Aberer et al. 2007 [9]; Jayasumana et al. 2007 [10]; Ciciriello et al. 2006 [11]; Kabadayi et al. 2006 [12]).

Due to the increased attention given to the subject of climate change, there is a rapidly rising demand for quality, high resolution gridded climatological datasets that provide detailed information on the variability of temperature and precipitation at regional scales. These data enable the spatially explicit investigation of complex near surface-atmosphere interactions over a larger, continuous region than the original climate station data permit

(Serbin et al. 2008 [13]) which will pave the way of continuous development in application of spatially explicit environmental models. Weather stations with long-term records and good spatial coverage are uncommon (Willmott et al. 1994 [14]); thus a challenge exists in gathering information from point data sources and generating from these a reasonable interpolation or extrapolation across topographically variable conditions at a variety of temporal and spatial scales. Generally, two problems arise from this lack of well-conditioned observational networks. Estimating a time-averaged weather or climate variable (e.g., air temperature) at unsampled locations by traditional spatial interpolation is relatively unreliable, and in turn, areal averages made from the network observations can be biased.

The uneven and relatively sparse distribution of stations that report daily climatological variables has led to rich literature describing and comparing techniques to interpolate these point measurements. There have been several studies that have applied simple spatial interpolation methods for spatial modeling of surface air temperature. Although each study has its own considerations, all traditional spatial interpolation methods that have been applied to air temperature mapping process can be categorized into one of the groups described below:

Inverse distance weighting (Willmott et al., 1985 [15]; Isaaks and Srivastava, 1989 [16]); The process of assigning temperature to unknown points by using values from scattered sets of known points (station locations with available temperature data). The unsampled points would be valued based on the simple weighting function which has an inverse relationship with distance from the known points.

Ordinary kriging (Deutsch & Journel 1992 [17]); In essence, as with inverse distance weighting, the estimation functions for ordinary kriging are a form of locally weighted averages in which the weights are derived following an initial investigation into the spatial structure of the data (variogram modeling). Kriging however, unlike inverse distance weighting, also takes into account the relative positions of the contributory sample data points. Moreover, the property being estimated (temperature) is treated as a regionalized variable whose variation over space can be modeled statistically.

Partial thin-plate splines (Hutchinson 1991 [18]; Jarvis & Stewart 2001a, b [19, 20]); Thin-plate splines are a physically based 2D interpolation scheme for arbitrarily spaced tabulated data $(x_i, y_i, f(x_i, y_i))$. These splines are the generalization of the natural cubic splines in 1D. The spline surface represents a thin metal sheet that is constrained not to move at the grid points. Smoothing splines allow both the smoothness and exactness of a surface fit to be considered together. The methodology of Hutchinson et al. 1991 [18] uses automatic generalized cross validation to optimize the smoothing function so that over-smoothing does not produce significant departures from the data points. In the 3D model, elevation is used as the third independent splining factor; in the 2D model, elevation is one of a number of partial linear covariates determined by the regression analyses.

Truncated Gaussian filtering (Thornton et al. 1997 [21]; Hasenauer et al. 2003 [22]); The process of using an iterative station density algorithm to determine a local set of predictor stations for each given control point, each with a weight that decreases with distance from the prediction point. Temperature gradients are calculated by summing the weighted contribution of the temperature gradient computed from every possible predictor station

pair, the interpolation weights are assigned to all stations in the database by using a truncated Gaussian filter which assigns weights according to a Gaussian probability density function that is truncated at some distance α from the mean.

Geostatistical methods (Jarvis & Stewart 2001a, b [19, 20]); Geostatistics is the study of phenomena that vary in space and/or time. Geostatistical methods use a semivariogram analysis to estimate the process using the best linear unbiased estimator. Jarvis et al. explored the derivation and selection of a comprehensive set of continuous topographic and land-cover influential guiding variables (e.g., North map coordinate, elevation, coastal and urban effects) to guide the interpolation of regional air temperature. These methods have the ability to update historical topoclimatic models with taking into account digital elevation and land cover data.

Cross-comparisons of above methods have revealed that simple interpolation is insufficient to capture the spatial variability of temperature (Stahl et al. 2006 [23]). Surface air temperatures are influenced not only by the macro-climatology of a site (i.e. regional patterns), but also by local features such as topography (e.g. cold air pooling), proximity to deep water bodies (e.g. ocean), proximity to high thermal capacity materials (e.g. urban heat island effect) (Lundquist et al. 2008 [24]; Jarvis & Stewart 2001a [19, 20]). Thus, it is necessary to develop spatial estimation methods that are more reliable than traditional interpolation methods. (Willmott et al. 1994 [14]).

Together with satellite observations, gridded meteorological variables can provide important information of the complex interactions of these features in order to accurately map temperature across broad regions. Particularly, satellite observations of land surface

temperature have been used successfully for urban landscape management (Quattrochi et al. 2000 [25]), urban environmental quality monitoring (Nichol & Wong, 2005 [26]), and urban risk analysis (Dousset et al. 2007 [27]). To gain knowledge about land surface temperature (LST) over urban areas from space, satellite-based sensors operating in the thermal infrared (TIR) spectral region are used as they provide LST image data at various spatial and temporal scales. However, due to technical constraints, satellite thermal sensors are unable to supply both spatially and temporally dense LST image data. The reason for this is that the spatial and temporal resolutions of a satellite thermal sensor are anti-correlated, meaning that a high spatial resolution is related with low temporal resolution and vice versa (Stathopoulou & Cartalis, 2009 [28]).

The trade-off between spatial and temporal resolution encouraged us to apply the Moderate Resolution Imaging Spectroradiometer (MODIS) as a source of remote-sensed land surface temperature data. MODIS is a multi-spectral imaging instrument that is carried aboard both the Terra and Aqua satellites; it takes measurements all day, every day; and it has a wide field view. MODIS's frequent coverage complements other imaging systems such as Landsat's Enhanced Thematic Mapper Plus, which reveals the earth in higher spatial details (Spatial Resolution: 30 m [60 m - thermal, 15-m pan]), but can only image a given area once every 16 days which is too infrequent to capture many of the rapid biological and meteorological changes that MODIS (Spatial Resolution [bands 20-23¹]: 1km, 5km) observes in every 1 to 2 days.

¹ MODIS has 36 different bands. Each band has specific bandwidth and primary usage. Bands 20-23 measure Surface/Cloud temperature.

1.2 Scope of Thesis

Through a review of the literature, and an analysis of existing ground-based temperature sensors, we have concluded that there are insufficient ground-based measurements to estimate the temperature throughout our study region. This implies that in general ground-based temperature measurements throughout the United States are too sparse to capture the spatial variability of temperature which makes obtaining sufficient data from ground-based sources to calibrate and validate temperature models difficult. Additionally, we have postulated that existing interpolation methods are insufficiently accurate to estimate the local temperature at ungauged locations because they cannot capture anthropogenic (e.g. urban heat island) or microclimatological (e.g. cold air pooling) effects. This work integrates both remote sensed and ground-based observations to account for anthropogenic and microclimatological impacts on the surface air temperature. Testing and demonstration of the spatial temperature model is accomplished using a case study from the State of New Jersey, which is in the Northeastern and Middle Atlantic regions of the United States with geo-location boundary box of Latitude $38^{\circ}56' N$ to $41^{\circ}21' N$, and Longitude $73^{\circ}54' W$ to $75^{\circ}34' W$, bordered on the north and east by the state of New York, on the southeast and south by the Atlantic Ocean on the west by Pennsylvania and on the southwest by Delaware. New Jersey contains high levels of urban development with the population density of $1,185/\text{mi}^2$ ($458/\text{km}^2$) which is ranked 1st in the US based on 2010 Census. Our method constructs continuous maps of temperature over our study region each day, and is able to capture the underlying spatial variability of temperature better than other traditional spatial methods.

This study is organized as follows. The next chapter (Chapter2) describes broad study of temperature data, and its availability over the state of New Jersey. Chapter3 introduces two popular conventional interpolation methods (Inverse distance weighting and truncated Gaussian filter) including variety of comparison tests, and demonstrating their performances, and limitations. The last chapter (Chapter4) introduces the application of remote sensing for purpose of temperature measurement, and describes the possible interpolation method which integrates both ground based, and remote sensed temperature data for purpose of temperature mapping.

Chapter2

Study of temperature data over the state of New Jersey

2.1 Introduction

Existing meteorological sensor networks are too sparse to accurately characterize the min/max temperature over broad spatial areas. Figure 1 illustrates the distribution of sensors participating in the National Climactic Data Center (NCDC's) Cooperative Observer Network over the state of New Jersey. Cooperative Stations are identified by a 6-digit number. The first two digits designate a USA state or territory code. The last four digits are assigned to stations within a state in general accordance with the alphabetic order of the station name (e.g., Atlantic City International Airport coop ID: 280311). It can be seen from figure 1 that the minimum, maximum, and average distance between sensors is 4.9 miles, 164.01miles, and 55.68 miles (Actual minimum distance is 2.63 miles between Sussex 2 NW and Sussex 8 NW, but Sussex 8 does not have data after May 2005), respectively. At the same time, the State of New Jersey has a land area of approximately 8700 square mile. Moreover, not all these existing sensors provide useful information for temperature estimation. For example, of the 41 Cooperative Observer

Network sensors that were operational between 1/1/98 and 1/1/2008, only 16 of them provide continuous temperature data over the desired period, resulting in an average station density of 1 station per 540 square miles.

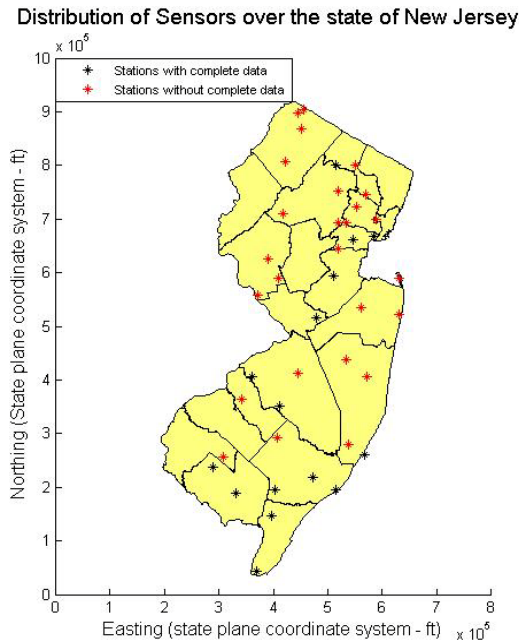


Figure 1 Distribution of sensors over the state of New Jersey

For better understanding of temperature data, the average maximum, minimum, and mean temperature for the study period (1/1/1998 – 1/1/2008) over the state of New Jersey based on historical stations data has been provided as the below.

Table 1 Average maximum, minimum, and mean temperature over the state of New Jersey (1/1/1995 – 1/1/2008)

1998		1999		2000		2001		2002		2003		2004		2005		2006		2007	
Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin	Tmax	Tmin
65.5	45.1	65.6	43.6	62.8	42.4	64.2	42.7	65.2	44.3	61.19	42.5	62.9	43.9	63.8	44.3	65.8	45.7	64.1	44.3
T mean																			
55.2717513		54.5813304		52.5588788		53.455134		54.7614751		51.8526685		53.3786364		54.0438706		55.758177		54.204786	

The availability of data in the current meteorological stations of New Jersey is presented in table 2 as below.

Table 2 Availability of data in New Jersey meteorological stations

Station Names	No Data Before	No Data After	Total Days without Data
ATLANTIC CITY			
ATLANTIC CITY INTL AP			
BELLEPLAIN S'IN FOREST			
BOONTON 1SE	1/1/2001		
BRANT BEACH BECH HAVEN			
CANOE BROOK			7
CAPEMAY 2 NW			
CHARLOTIESBURG RS VR			
CHATHAM 2W		1/1/2004	
CRANFORD			
ESSEX FELS SVC BLDG			108
ESTELL MANOR			
FLEMINGTON 5 NNW			55
FREEHOLD-MARLBORO			61
HAMMONTON 1 NE	1/1/2004		
HARRISON	1/1/2001		33
HIGH POINT PARK	1/1/2000	5/31/2005	
HIGHTOWN 2W			
INDIAN MILLS 2W			
LAKEHURST NAS			There is no 7 consecutive days with data
LAMBERTVILLE		2/1/2002	42
LITTLE FALLS		2/29/2004	10
LONG BRANCH OAKHURST		11/30/2006	70
LONG VALLEY		9/22/2004	82
MILLVILLE MUNI AP			
MOORESTOWN			
NEW BRUNSWICK 3 SE			
NEWARK INTL AP			
NEWTON		9/30/2004	19
PEMBERTON		2/28/2002	
PLAINFIELD			59
SANDY HOOK			600
SEABROOK FARMS			
SOMERVILLE 4NW	2/1/2000	5/31/2006	
SUSSEX 2 NW			183
SUSSEX 8 NW		5/31/2005	
TOMS RIVER			26
TUCKERTON 2 NE	11/1/2004		
WANAQUE RAYMOND DAM		1/8/2004	108
WERTSVILLE 4 NE			23

2.2 Seasonality Removal

In statistics, many time series exhibit cyclic variation known as seasonality, periodic variation, or periodic fluctuations. This variation can be either regular or semiregular. Initial investigation of the temperature data revealed a significant temporal trend, which is illustrated in Figure 2. This temporal trend is due to the seasonal fluctuation from coldest (mid-January) to warmest (early August).

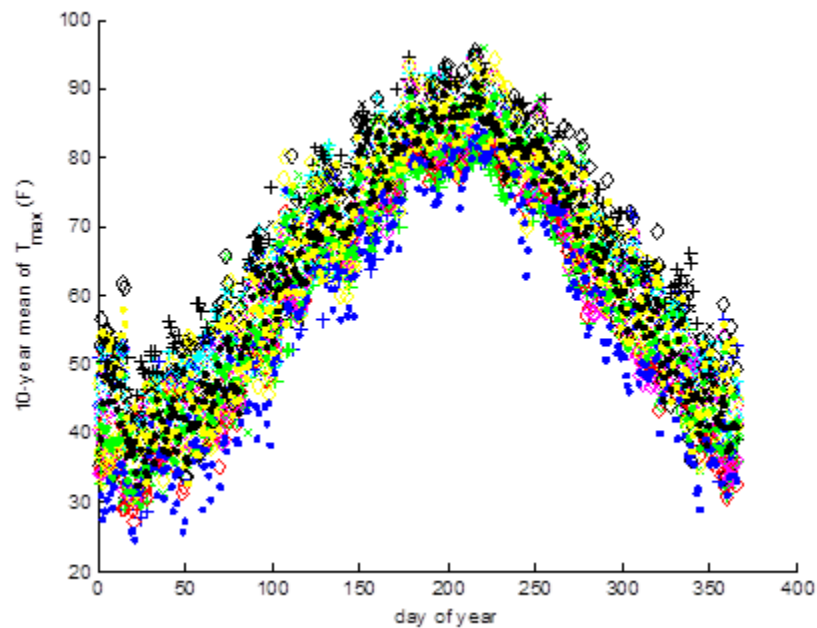


Figure 2 Ten-year average of maximum daily temperature at each cooperative network sensor location

2.2.1 Statistical analysis for existence of seasonality

Autocorrelation refers to the correlation of a time series with its own past and future values. Informally, it is the similarity between observations as a function of the time separation between them. It is a mathematical tool for finding repeating patterns such as seasonality, and detecting non-randomness in data. Autocorrelation plot axes definitions are as below:

- Vertical axis: autocorrelation coefficient:

$$R_h = C_h / C_0$$

(1)

$$C_h = \frac{1}{N} \sum_{t=1}^{N-h} [(X_t - \bar{X})(X_{t-h} - \bar{X})]$$

(2)

$$C_0 = \frac{1}{N} \sum_{t=1}^N [(X_t - \bar{X})^2]$$

(3)

- Horizontal axis: lag

The autocorrelation plot for ten-year of maximum daily temperature is demonstrated as below:

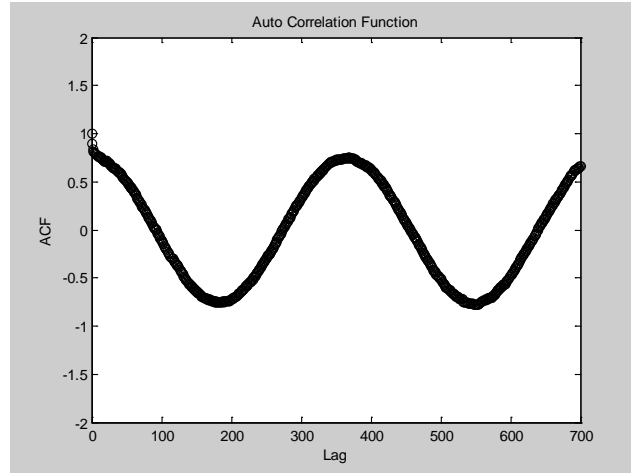


Figure 3 Autocorrelation plot for ten-year of maximum temperature

Autocorrelation structure of a time stationary process will not change over time. The oscillation of the autocorrelation plot will confirm the existence of seasonal trend. Moreover, because the autocorrelation plot does not decay to zero the non-randomness of data will be confirmed indeed. Using available data, we modeled this seasonal trend as a series of sine terms:

$$T_{max}(t) = 63.7004 - 20.7831 \cos(0.0172t) - 8.1297 \sin(0.0172t) \quad (4)$$

$$T_{min}(t) = 44.8093 - 19.2716 \cos(0.0172t) - 8.1801 \sin(0.0172t) \quad (5)$$

Subtracting 10-year maximum temperature from the trend-equation T_{max} gives the detrended maximum daily temperature data, shown in figure 4 (figure 4 is the detrended version of figure 2). As it can be seen in this figure the detrended data do not exhibit a temporal trend.

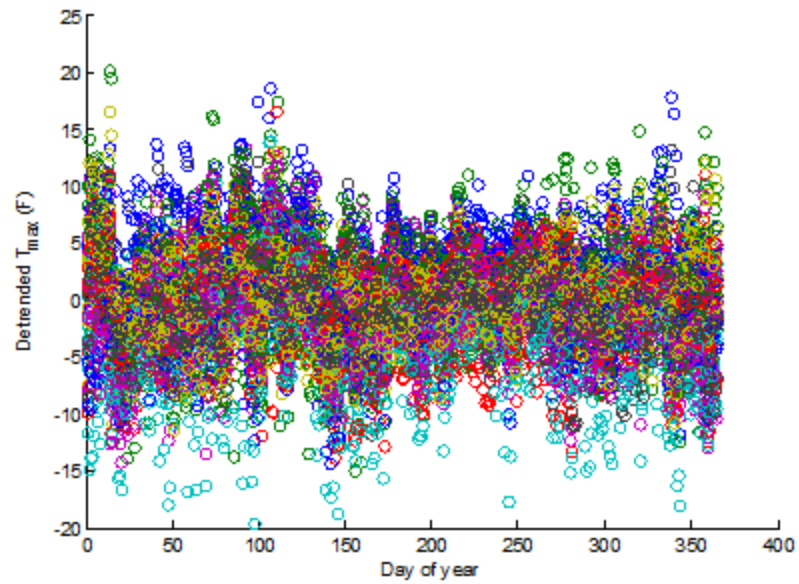


Figure 4 De-trended ten-year average maximum temperature at each cooperative network sensor location

To validate the removal of seasonality, partial autocorrelation plot has been provided.

Partial autocorrelation plots are formed by:

- Vertical axis: Partial autocorrelation coefficient at lag h .
- Horizontal axis: Time lag h ($h = 0, 1, 2, 3 \dots$).

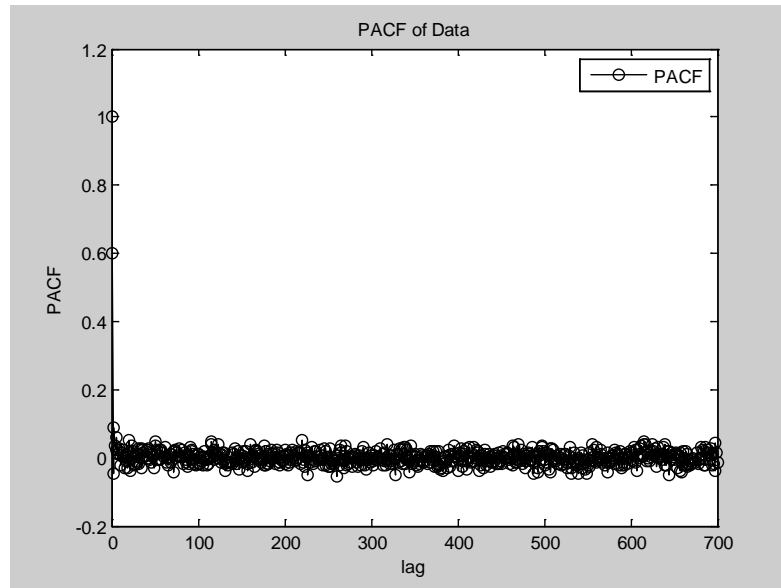


Figure 5 Partial autocorrelation plot for ten-year maximum temperature

It can be clearly seen (figure 4), and statistically approved (figure 5) that the seasonal trend has been removed successfully.

2.3 Semivariogram Analysis

Geostatistics is the study of phenomena that vary in space and/or time (Deutch, 2002 [29]). “It can be regarded as a collection of numerical techniques that deal with the characterization of spatial attributes, employing primarily random models in a manner similar to the way in which time series analysis characterizes temporal data.” (Isaaks and Srivastava, 1989 [16]). One of the basic components of geostatistics for characterization of spatial correlation is the semivariogram analysis.

To explore the ability of the existing Cooperative Observer Network sensors to characterize the spatial variability of temperature, a semivariogram analysis was performed. The semivariance can be computed using the following formula:

$$\gamma = \frac{1}{2N(h)} \sum_{\alpha=1}^{N(h)} (z(u_{\alpha}) - z(u_{\alpha} + h))^2 \quad (6)$$

Where $N(h)$ are the number of data pairs separated by a distance h , $z(u_{\alpha})$ is the measurement of the value of interest at location u_{α} , and $z(u_{\alpha} + h)$ is the lagged measurement of the value of interest at location $u_{\alpha} + h$. Sometimes $z(u_{\alpha})$ will be referred to as the “tail” variable and $z(u_{\alpha} + h)$ will be referred to as the “head” variable. Thus, from Equation 6, the semivariance at distance h is equivalent to half the average squared difference between measurements separated by distance h . If sufficient information exists to characterize the small scale variability of the phenomenon of interest, the semivariogram will be approximately zero at the smallest spatial separation distance. Furthermore, if there are no significant spatial trends, the semivariance will asymptotically approach the global inter-station covariance which has been demonstrated as below:

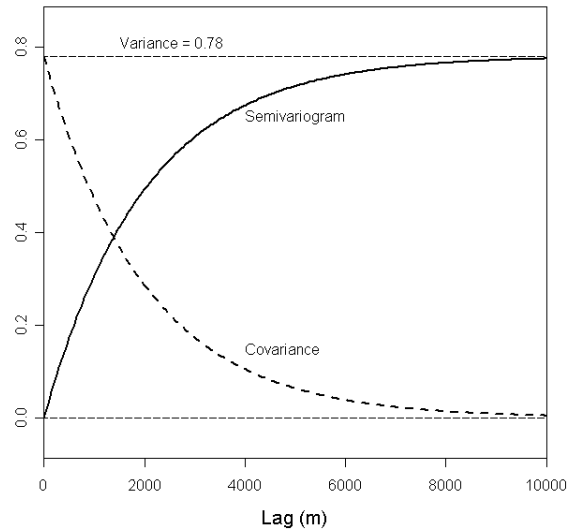


Figure 6 Sample semivariogram of data without significant spatial trends

The isotropic (i.e. direction independent) semivariogram computed from the measurements made by the Cooperative Network sensors between 1/1/1998 and 1/1/2008 is shown in Figure 7. Since it is well-known that a temporal trend exists in the data, the temporal trend was removed before computing the semivariance. Figure 7 clearly shows a significant deviation from zero near the origin. This implies that the stations are too sparse to characterize the small scale variability of the temperature. Additionally, because the semivariance grows without bound, it can be determined that there are significant spatial trends within the data.

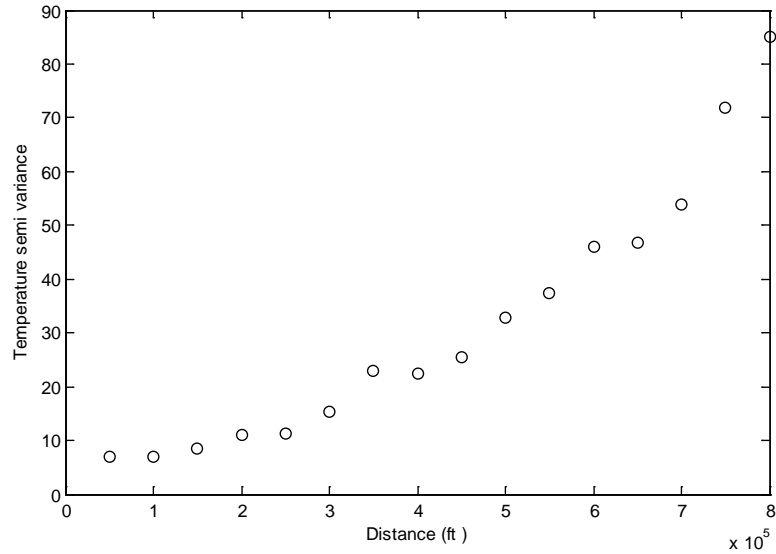


Figure 7 The semivariogram plot for sensors

2.4 Temperature modeling challenges

Since our exploratory analysis indicated that there are significant spatial and temporal trends in the surface air temperature, our model needs to capture this trend. Our approach decouples these trends. The temporal trend is described by Equation 4, whereas a model of the spatial trend must still be developed since prior studies have shown that straightforward applications of interpolation methods will not provide accurate results when there is a significant spatial trend in the data. Thus, we have explored several anthropogenic and microclimatologic drivers suggested in the literature, namely, northing, elevation, and log population (Jarvis et al. 2001 [20]). To identify these trends, we investigated plots of temperature versus driving feature. Figure 8, 9, 10, 11, 12 and 13 show examples of such plots for temperature vs. elevation, temperature vs. log population, and temperature vs. northing. Again to increase the generality of the result, the temporal trend was removed using Equation 4.

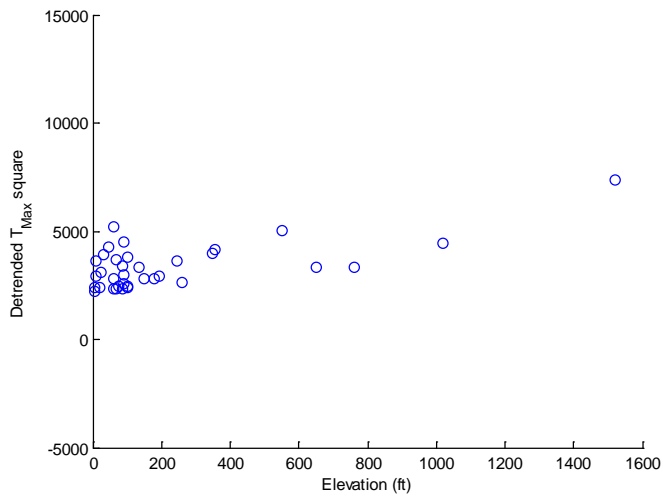


Figure 8 De-trended maximum temperature square versus elevation

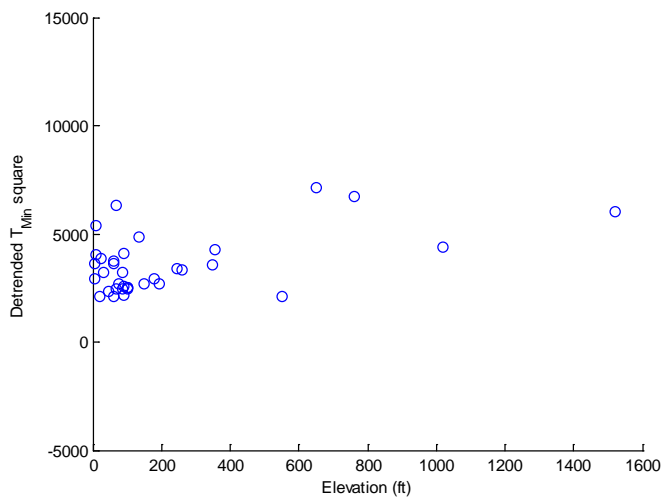


Figure 9 De-trended minimum temperature versus elevation

It should be mentioned here that although in lots of spatial interpolation studies lapse rate has been used for adjusting the temperature for different altitudes ([21, 30, 31, 32, 33, 34, 35, 36, 37,38]), but the above figures (Figure8, and 9) show that there is not a clear

trend between temperature and elevation. In general, a lapse rate is the negative of the rate of temperature change with altitude change with mathematical definition of:

$$\gamma = -dT/dz \tag{7}$$

Where γ is the lapse rate given in units of temperature (T) divided by units of altitude (z).

Plots for temperature vs. log population, and temperature vs. northing:

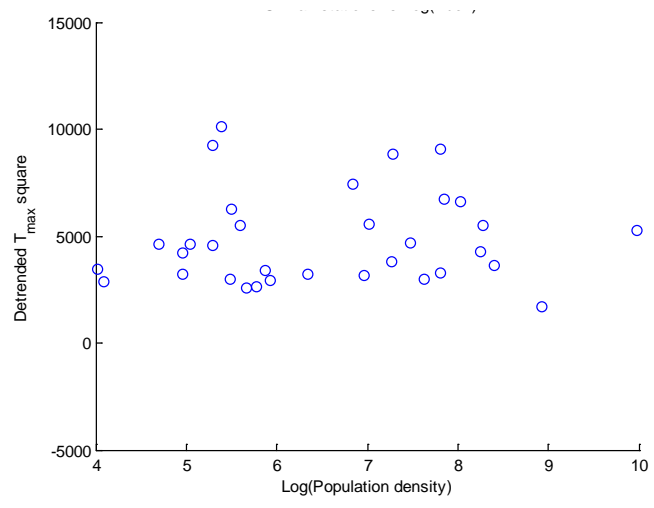


Figure 10 De-trended maximum temperature versus log of population density

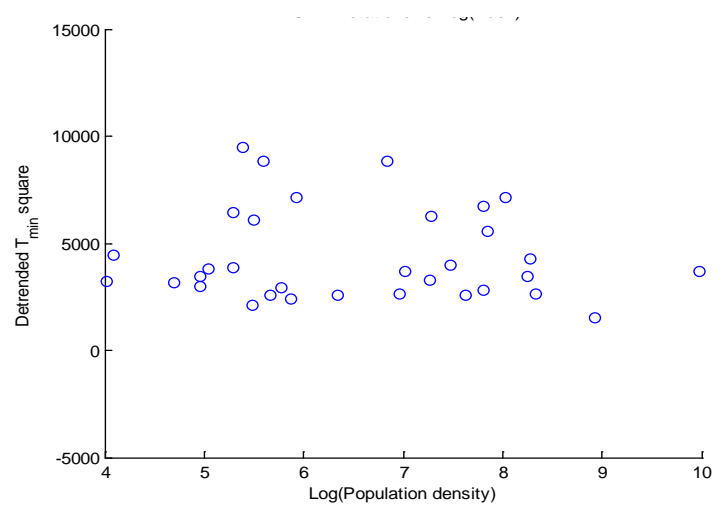


Figure 11 De-trended minimum temperature versus log of population density

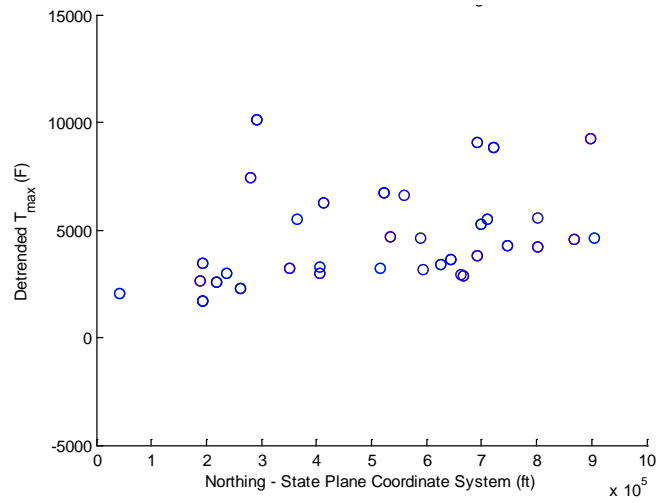


Figure 12 De-trended maximum temperature versus northing

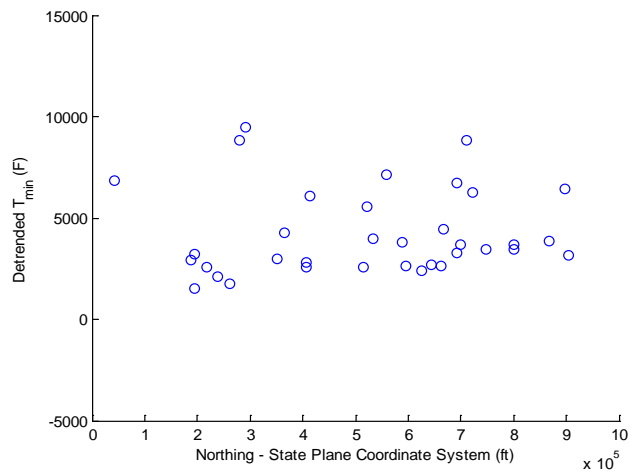


Figure 13 De-trended minimum temperature versus northing

2.5 Conclusion

Despite the obvious existence of a trend illustrated by the semivariogram analysis, the pairwise correlation analysis illustrated in Figures 8, 9, 10, 11, 12, and 13 does not exhibit a clear spatial trend in the data. We suspect that result is caused by 2 primary features of the system: (1) insufficient data density and (2) complex interactions of the temperature drivers. Since there are fewer than 20 stations providing continuous temperature data throughout the State of New Jersey, there may be insufficient data to characterize the relationship between local temperature and ancillary anthropogenic and microclimatologic features. However, it is more likely that the complex interaction of these drivers is obscuring the relationships. New Jersey is the most heavily urbanized state in the US, and it is proximal to the 1st (New York City) and 6th (Philadelphia) largest cities in the US. At the same time there is significant geologic variation from the mountainous northwestern region of NJ to the Pine Barrens in the south. New Jersey also has a significant amount of Atlantic Ocean coastline, and has several major rivers (Delaware, Hudson, and Raritan). Thus, we expect that there are multiple strong influences on the data from these drivers.

Chapter 3

Inverse Distance Weighting vs. Truncated Gaussian Filter

3.1 Introduction

Most of spatial interpolation models apply weighted average models, and they vary with each other in weight calculation method. Two popular weight calculation methods are Inverse Distance, and Gaussian Filter. For this reason two interpolation procedures include of Inverse Distance Weighting and Truncated Gaussian filter are presented and assessed with respect to their ability to estimate maximum and minimum land surface temperature at unsampled points from available station data. To test the possibility of getting better results over each method, truncated Gaussian filter (TGF) (Thornton et al. 1997 [21]) and inverse distance weighting (IDW) (Isaaks & Srivastava, 1989 [16]) implemented in the MATLAB computing language. Brief introduction of each method is provided as below.

3.1.1 Inverse Distance Weighting (IDW)

Inverse distance weighting (IDW) is a method for multivariate interpolation, a process of assigning values to unknown points by using values from usually scattered set of known points. Here, the value at the unknown point is a weighted sum of the values of N known points. A general form of finding an interpolated value u at a given point x based on samples $u_i = u(x_i)$ for $i = 0, 1, \dots, N$ using IDW is an interpolating function:

$$u(x) = \frac{\sum_{i=0}^N \frac{w_i(x)u_i}{\sum_{j=0}^N w_j(x)}}{\sum_{j=0}^N w_j(x)} \quad (8)$$

$$\text{Where: } w_i(x) = \frac{1}{d(x, x_i)^p} \quad (9)$$

x denotes an interpolated (arbitrary) point, x_i is an interpolating (known) point, d is a given distance (metric operator) from the known point x_i to the unknown point x , N is the total number of known points used in interpolation and p is a positive real number, called the power parameter representing the decay in similarity between values over distance (Jarvis et al. 2000 [19, 20]).

3.1.2 Truncated Gaussian Filter (TGF)

The general form of the truncated Gaussian filter, with respect to its central point, p , is

$$w(r) = \begin{cases} 0 & , r > R_p \\ \exp [-(r/R_p)^2 \alpha] - e^{-\alpha} & , r \leq R_p \end{cases} \quad (10)$$

Where $w(r)$ is the filter weight associated with a radial distance r from p , R_p is the truncation distance from p , and α is a unit-less shape parameter.

The spatial convolution of this filter with a set of horizontal station locations results, for each point of prediction, in a list of weights associated with observations. Because of the spatially heterogeneous distribution of observations, a constant value for R_p results in a large disparity in the number of observations with non-zero weights between points in the least and the most densely populated regions of the prediction grid. R_p can be reduced in data-rich regions, using information from a smaller radius, and increased in data-poor regions. For this purpose TGF specifies N , the average number of observations to be included at each point. R_p is then varied as a smooth function of the local station density in such a way that this average is achieved over the spatial domain. The smooth variation of R_p ensures a continuous interpolation surface, and is accomplished through the iterative estimation of local station density at each prediction point, as follows:

- I. For all grid cells, the same user-specified value, R , is used to initialize R_p .
2. Given R_p , equation (10) is used to calculate weights w_i , where $i = (1, \dots, n)$ are observation locations, and the local station density D_p (number of stations/area) is then determined

$$D_p = \frac{\sum_{i=1}^n \frac{w_i}{\bar{w}}}{\pi R_p^2} \quad (11)$$

Where \bar{w} is the average weight over the untruncated region of the kernel, defined as

$$\bar{w} = \int_0^{R_p} w(r) \cdot dr / \pi R_p^2 = (1 - e^{-\alpha} / \alpha) - e^{-\alpha} \quad (12)$$

3. A new R_p is calculated as a function of the desired average number of observations, N , and the most recent calculation of D_p as:

$$R_p = \sqrt{N^* / D_p \cdot \pi} \quad (13)$$

$N^* = 2N$, for the first $I - 1$ iterations

$N^* = N$, final iteration

This modification of N^* is a result of filter truncation, and helps to avoid the occasional occurrence of excessively large station counts in regions of strongly heterogeneous station spacing.

4. The new R_p is substituted in Step (2), and Steps (2)-(4) are iterated a specified number of times, I . Final values of R_p are incorporated in Eq. (10) to generate the interpolation weights W_i used in predictions for all days at the point in question.

The interpolation method for a given set of observations and a given prediction grid is defined by the four parameters R , I , N , and a . Given an arbitrary variable x_i , measured at each of the $i = (1, \dots, n)$ observation points, values for the interpolation parameters are specified once and held constant over all days and all prediction points. Taking the case of a single prediction point on a single day, the interpolated value x_p is determined in general

$$x_p = \frac{\sum_{i=1}^n W_i x_i}{\sum_{i=1}^n W_i} \quad (14)$$

3.2 Interpolation Implementation

Our implementation of these methods relied on several internal MATLAB functions. The following gives a brief explanation of these functions followed by an explanation of our implementation.

For generating uniform random numbers, two functions have been used:

- Unidrnd(N), which generates random numbers for the discrete uniform distribution with maximum N.
- Unifrnd(A,B) returns an array R of random numbers generated from the continuous uniform distributions with lower and upper endpoints specified by A and B, respectively.

The interpolation started over plain surface using random observation (sensor) locations defining random value for each observation point. Unidrnd(N) MATLAB function is used to construct our locations, while unifrnd(-1,1) built normally distributed value between -1 and 1 for each of observation points. IDW acts better in this circumstances ($MSE_{IDW}=0.47$, $MSE_{TGF}=0.6$), but the values of our sensors don't have any spatial correlations with each other. In other words, there was no discernable pattern between values with keeping in mind that the very first concept of all interpolation methods is: the objects which are closer to each other, are more alike. This idea encouraged us to change the testing criteria to a more realistic fashion. This task was performed by adding

Gaussian distributions to the plain surface with MATLAB peaks function (Peaks is a function of two variables, obtained by translating and scaling Gaussian distributions. Peaks can be defined as a mixture of 3-D Gaussian distributions) in order to get functional spatial distribution. The conventional Peaks function is demonstrated as below:

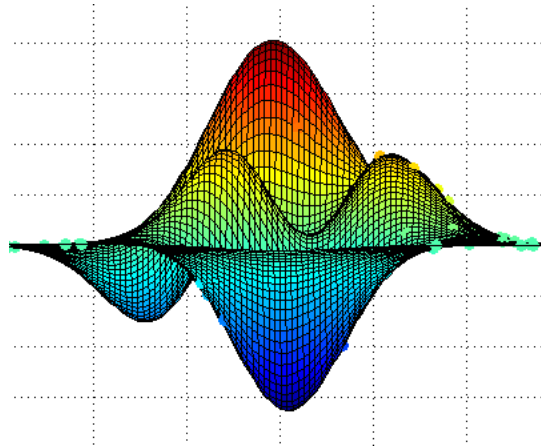


Figure 14 Conventional peaks function

3.3 Test validation procedure: Mean Square Error (MSE)

To evaluate comparatively the performance of each interpolation method in each testing circumstances we employed mean square error. The algorithm of MSE is described as below:

$$MSE_{IDW \text{ or } TGF} = \frac{1}{n} \sum_{i=1}^n (Residuals)^2 \quad (15)$$

$$Residuals = Interpolated \ Surface - Actual \ Surface$$

While the sensors locations are random with random values each time, the delivered MSE value in each run cannot identified as a good judgmental tool because of wide tolerance which is demonstrated as below while dealing with 10 sensors.

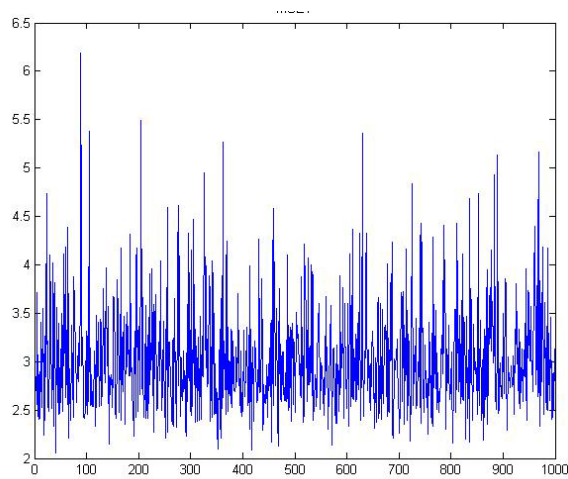


Figure 15 MSE distribution TGF (10 sensors included)

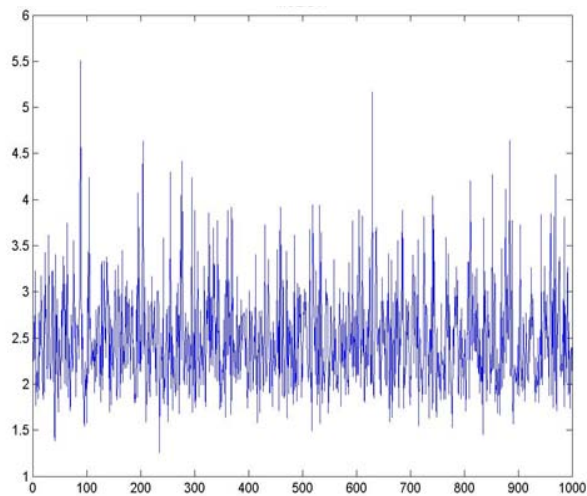


Figure 16 MSE distribution IDW (10 sensors included)

The solution of reaching to the constant value of MSE can be provided by running our models in big significant trials until finding statistics convergence, in such a way that MSE in each trial will be added to the previous MSE values, and then the summation will be averaged.

3.4 Parameterizing of Models Over Peaks Data

3.4.1 Truncated Gaussian Filter Parameter Justification

TGF is so sensitive to its parameters such as: Shape parameter (α), truncation radius (R_p) and, number of iterations (i), and in each different criteria it needs precisely selected parameters dependent on the underlying data pattern which should be obtained by trial and error. (Thornton et al. 1997 [21]). We applied parameterization of the TGF model for the peaks data by a process in which one parameter was varied while the rest were held constant. The value of the varied parameter that minimized the MSE was selected.

3.4.2 Inverse Distance Weighting Parameter Justification

IDW's parameter which should be justified is number of included neighbors (P) which indicates that it could have more robust functionality than TGF in the real world.

Table 3 displays the result summery of our trial and errors for both models in different criteria. The most left cells indicate the parameters. The parameters which remained

constant are shown in red-colored cells. The yellow highlighted band shows the best achieved results which were applied as our TGF parameters.

Table 3 Result summary of parameter justifications of IDW and TGF

		Mean Square Error (MSE)											
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
		attempt1	attempt2		attempt3		attempt4		attempt5		average		
$\rho=5$	20	6.5	5.77	5.9	5.5	2	1.38	2.3	2.4	4.42	4.41	4.224	3.892
$\alpha=2$	30	4.83	4.24	4.38	4	5.47	4.53	2.91	2.4	2	1.66	3.918	3.366
$l=3$	40	1.78	2	1.37	1.48	2.98	2.81	2.76	2.39	6.69	6.33	3.116	3.002
$R=3$	50	6.95	6.3	4.2	4.4	5.21	5.19	4.67	4.58	4.98	4.67	5.202	5.028
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	2.48	2.26	6.43	6.8	1.38	1.18	2.18	2	4.57	4.6	3.408	3.368
$\alpha=2$	30	3.12	2.55	5.2	4.6	4	4	2.2	2.2	2.37	2.5	3.378	3.17
$l=3$	40	2.3	2.1	3.55	3.6	5.3	4.2	4	3	1.4	1.4	3.31	2.86
$R=3$	50	2.9	2.17	3	3	2.2	2	4.8	4.4	4.6	4	3.5	3.114
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	2.18	2.1	1.08	1.1	1.6	1.5	3.2	3	1.64	1.5	1.94	1.84
$\alpha=2$	30	3.3	3	4.2	3.7	3.3	3.2	1.76	1.74	3.46	3.3	3.204	2.988
$l=3$	40												
$R=2$	50												
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	2.2	2	3.64	3.77	6.21	6.55	2.14	1.85	2.49	2.69	3.336	3.372
$\alpha=2$	30												
$l=3$	40												
$R=1$	50												
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	5.83	4.95	4.76	4.45	3.5	3.2	5.8	5.4	3.2	3.2	4.618	4.24
$\alpha=2$	30												
$l=4$	40												
$R=2$	50												
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	5.5	4.6	1.2	1.19	1.5	1.4	0.56	0.54	3.2	3.8	2.392	2.306
$\alpha=2$	30	3.43	3.2	4.52	4.3	3.3	3.2	1.9	1.8	4.4	3.1	3.51	3.12
$l=5$	40												
$R=2$	50												
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	2.47	2	5.2	3.8	1.63	1.63	1.1	1.08	0.86	0.73	2.252	1.848
$\alpha=1$	30	3.5	3.7	1.37	1.23	4.3	4.3	3.2	2.8	3.6	3.2	3.194	3.046
$l=3$	40												
$R=2$	50												
Nsample		IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF	IDW	TGF
$\rho=4$	20	3.4	3.2	4.7	4.5	4.4	4.4	2.57	2.54	4.1	3.7	3.834	3.668
$\alpha=3$	30												
$l=3$	40												
$R=2$	50												

3.5 Test of Interpolation Performance

After parameters justifications the interpolation models were ran. In each step of running the number of sensors was increased for sake of visioning the possible privilege of either

model based on available sensors number. In each run, the sensor values were assigned randomly from the continuous uniform distribution between $[-1, 1]$ as well as sensor locations which were assigned randomly from discrete uniform distribution. The following number of sensors was applied: 10, 20, 40, 80, 160, 320, and 640 respectively.

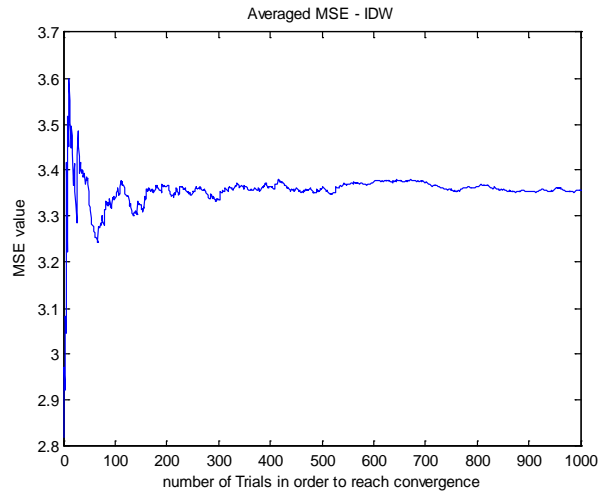


Figure 17 MSE value IDW Number of sensors: 10

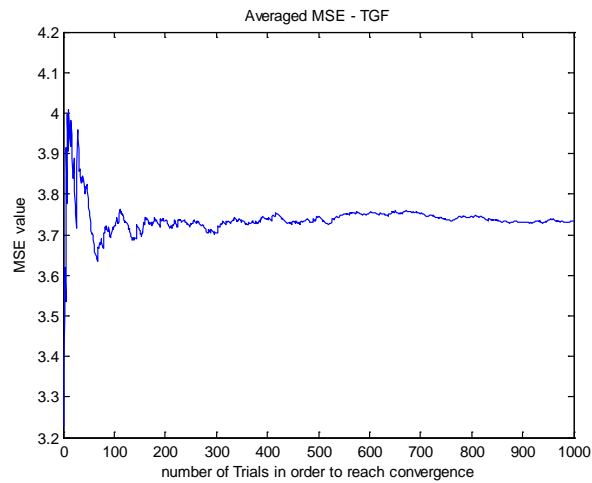


Figure 18 MSE vale TGF Number of sensors: 10

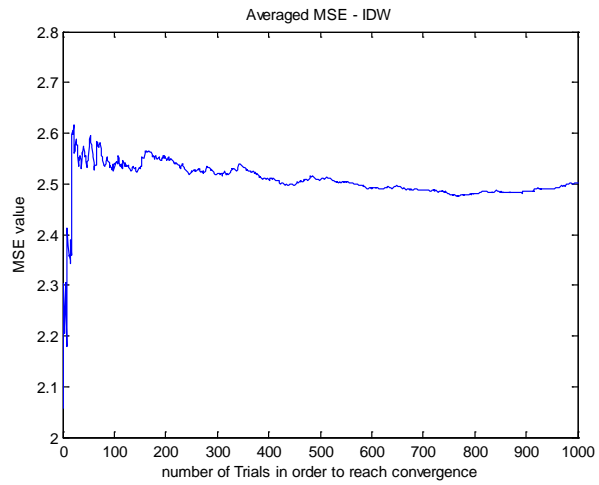


Figure 19 MSE value IDW, Number of sensors: 20

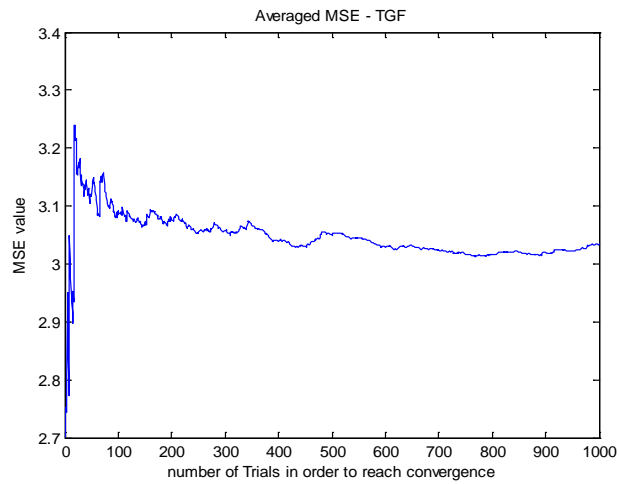


Figure 20 MSE value TGF, Number of sensors: 20

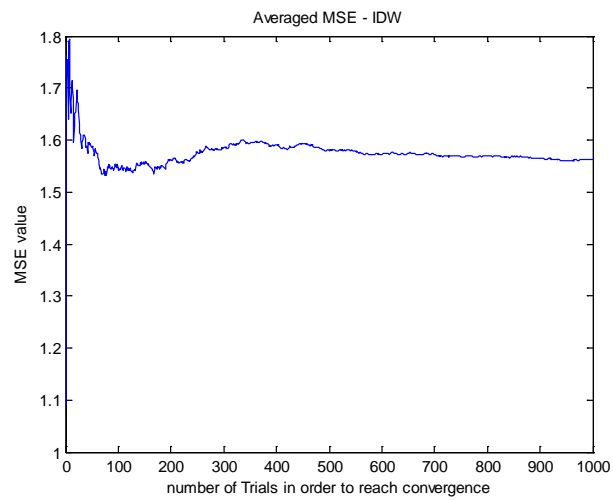


Figure 21 MSE value IDW, Number of sensors: 40

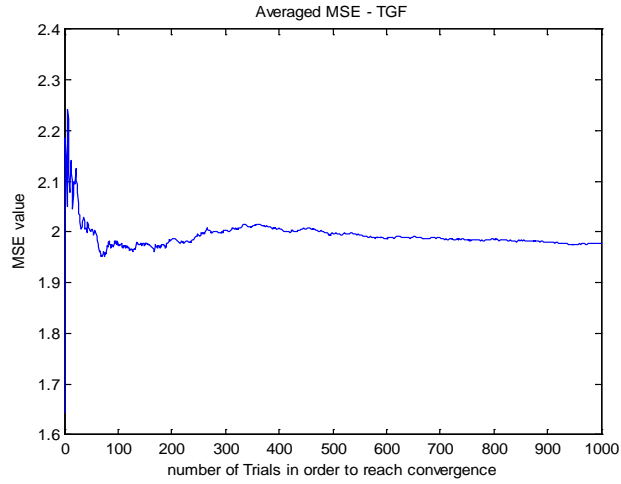


Figure 22 MSE value TGF, Number of sensors: 40

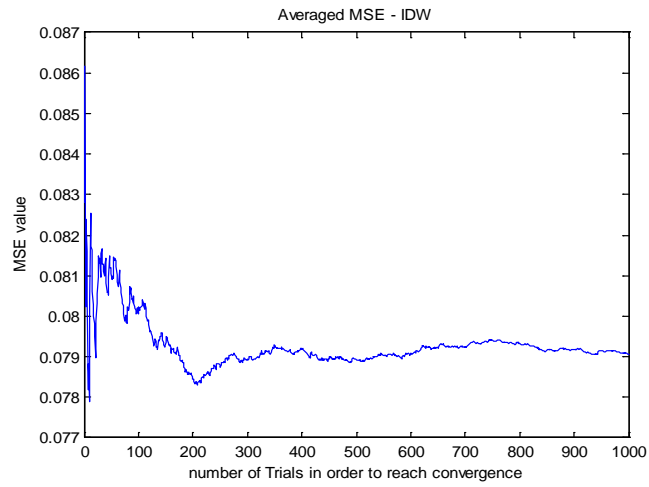


Figure 23 MSE value IDW, Number of sensors: 640

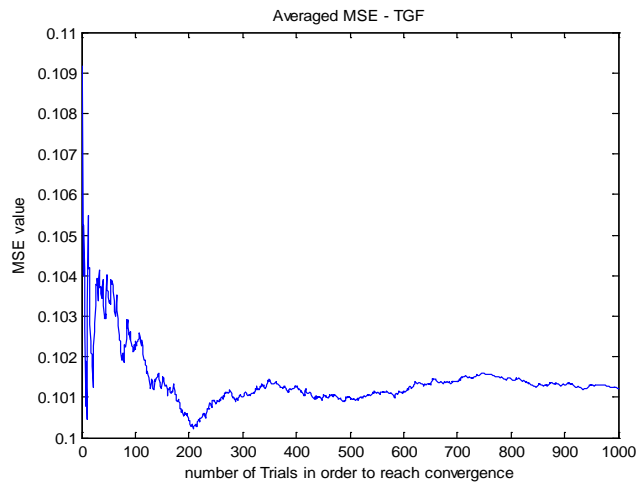


Figure 24 MSE value TGF, Number of sensors: 640

Figures 25 and 26 demonstrate sensor values versus interpolated values via TGF and IDW methods. Red lines represent sensor values, and Black lines represent interpolated values, and Table 4 provided the summary of MSE results. The fact that black line from IDW method follows red line better than that of TGF together with less IDW's MSE than TGF's MSE, indicates that IDW does slightly better job with considering that TGF interpolation takes much more time than IDW due to its iteration procedure. Moreover, in each criterion TGF needs new parameter adjustments.

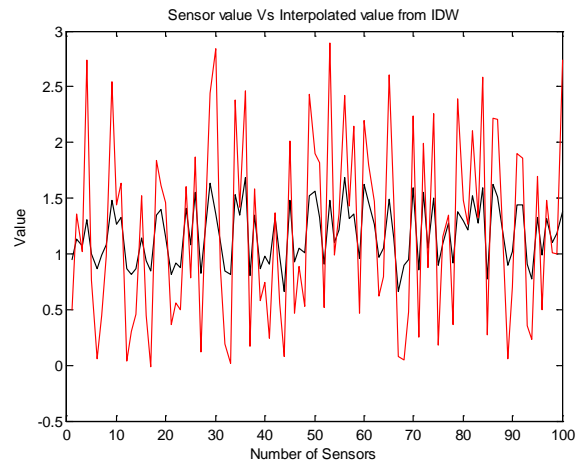


Figure 25 Sensor value Vs. interpolated value from IDW

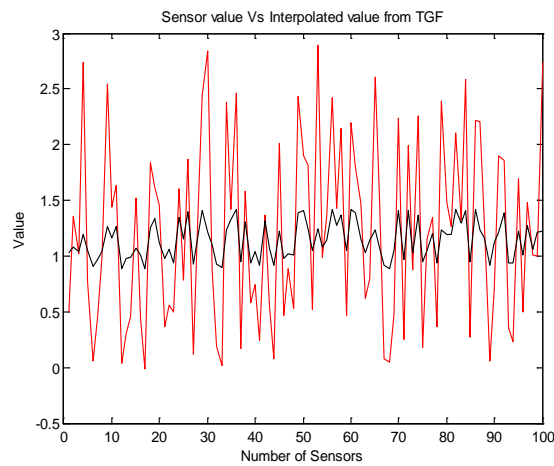


Figure 26 Sensor value Vs. interpolated value from TGF

Table 4 MSE values IDW and TGF

Number of sensors	MSE		Computation time	
	IDW	TGF	IDW	TGF
10	2.48	3.01		
20	2.48	3.01		
40	1.56	1.97		
80	0.8	1.04	<10sec	<1800sec
160	0.7988	1.039	<10sec	<3600sec
320	0.7986	1.039	<70sec	8000sec
640	0.079	0.101	<120sec	10800sec

3.6 Defining a More Realistic Surface

Working with the peaks function presented some challenges. The peaks function is too peaky in some parts, and much of the surface values approximately equal to zero. In fact, spatial data patterns like this are rare in rapidly mixed environment such as the near-earth atmosphere (although they can be found in the subsurface or in rainfall patterns). For this reason we came up with an idea of constructing smoother surface which consists of four Gaussian distributions with the ability to changing the smoothness of our peaks with increasing the standard deviation of each distribution. In fact heights of each peak are under control by changing mean value of each distribution. The new surface is illustrated as below.

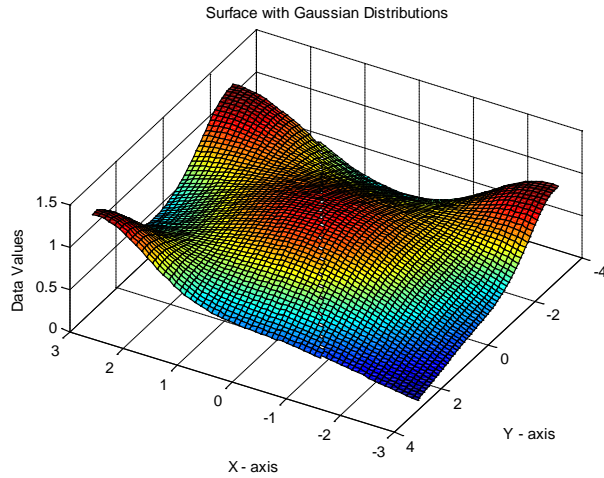


Figure 27 Surface with Gaussian distributions

Surface with smooth Gaussian distributions can examine models behavior in a more realistic condition. Because this study purpose is to observing any possible privilege of either model, simple inclined plane was also added to our comparison field for sake of observing the methods behavior in this contrived circumstance. The sample inclined surface can be seen in the following figure.

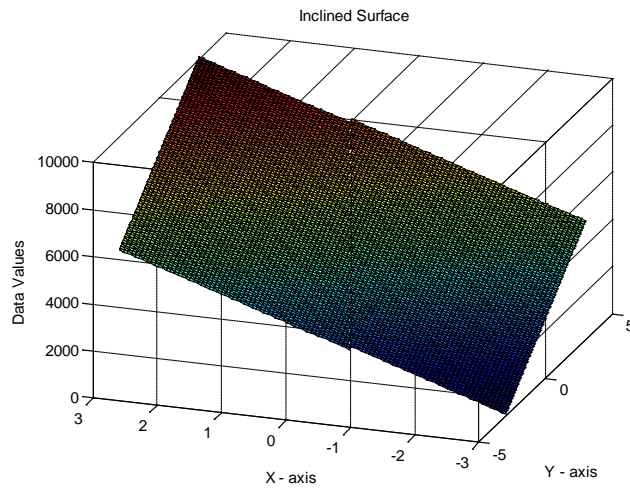


Figure 28 Inclined surface

3.7 Test of Interpolation over regions with “Sparse & Clusters” of sensors

One of the stated benefits of TGF over the rest of interpolation methods is the better results in situations of dealing with clusters of sensors in some points and sparse number of sensors in the whole area (Thornton et al. 1997 [21]). For the purpose of comparing our two models in these circumstances we defined areas with cluster of sensors by a square which contains 8 sensors. The sample field is demonstrated in the below image. In this step of experiment we are comparing TGF with IDW when the sensor distribution contains both clusters and sparse locations while the number of clusters & sparse points will increase in each run.

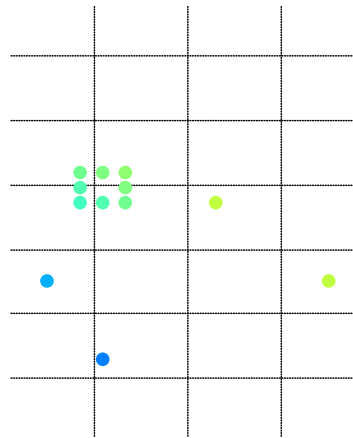


Figure 29 Sparse and cluster distribution of sensors

Table 5 Shows the calculated MSE over two surfaces while the number of clusters is increasing (There is no relationship between magnitudes of MSE in two surfaces because the range of data values [z-axis] is different in each criterion, and the idea was just comparing values over two methods). The result of this test indicates, however TGF acts

better than IDW for prediction of cluster centers, but the overall performance of IDW over whole surfaces is slightly more accurate than TGF.

Table 5 MSE values for inclined surface and Gaussian surface

Inclined Surface - Center of Cluster actual value Vs. Interpolated Value					Gaussian Surface - Center of Cluster actual value Vs. Interpolated Value				
4points+1cluster				MSE	4points+1cluster				MSE
1					1				
TGF	5000			7.1	TGF	0.71			0.049
IDW	5115			7.3	IDW	0.7			0.0505
Actual value	5000				Actual value	0.7			
8points + 2cluster				MSE	8points + 2cluster				MSE
1		2			1		2		
TGF	5000	2604		7	TGF	0.71	0.512		0.04
IDW	5115	2685		2.4	IDW	0.7	0.4732		0.02
Actual value	5000	2600			Actual value	0.7	0.4976		
12points + 3cluster				MSE	12points + 3cluster				MSE
1	2		3		1	2		3	
TGF	5000	2604	3107	6.8	TGF	0.71	0.512	1	0.039
IDW	5115	2685	3080	1.9	IDW	0.7	0.4732	0.997	0.018
Actual value	5000	2600	3100		Actual value	0.7	0.4976	1	

3.8 Defining the number of nearest neighbors in IDW

In the IDW method defining the number of nearest neighbors will affect the quality of interpolator. The conventional IDW method uses constant number of nearest neighbors which should be defined as an input. IDW can be modified in such a way that the number of nearest neighbors which are involved in the interpolation procedure, vary based on the

availability of sensors. For better illustration of the conventional IDW and modified version, the general flowchart has been demonstrated as below, where $R_{p,i}$ is the distance between point of interest and neighbor point i .

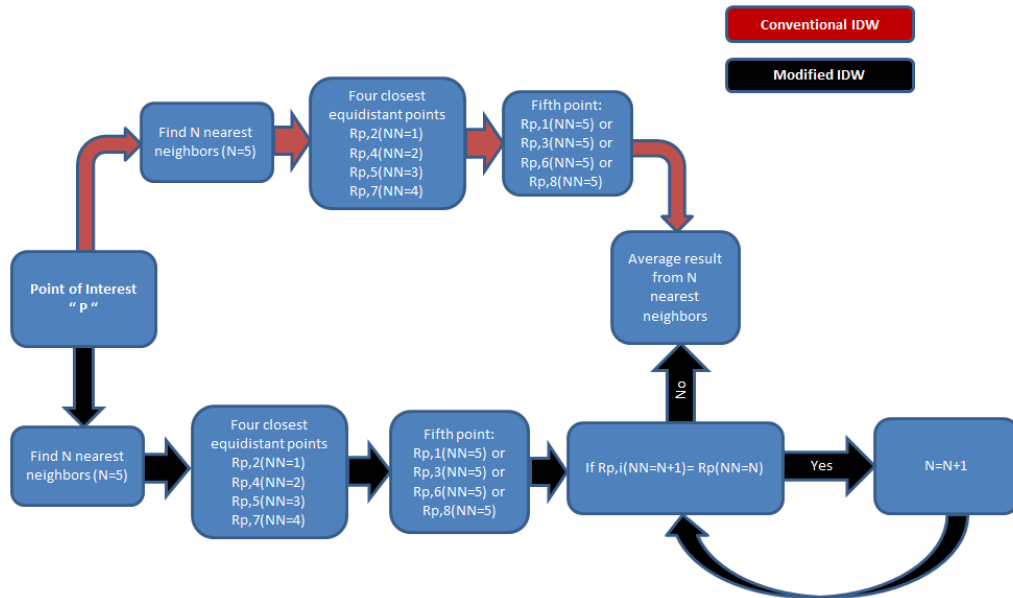


Figure 30 IDW and modified-IDW flowchart

Here, the difference between IDW and the modified version will be discussed through the specific condition. In the sample cluster situation which has been demonstrated in fig31² if the number of nearest neighbors is set by five, the model will miss three observation points. IDW acts like this: first it picks four closer equidistant points 2, 4, 5, and 7. for the fifth nearest neighbor it randomly picks one of the equidistant points 1, 3, 6 or 8. Modifying IDW such that all four of the points 1, 3, 6, and 8 are included in the interpolator is a more logical strategy than picking only one of them at random.

² 2,4,5, and 7 are equidistant
1,3,6, and 8 are equidistant

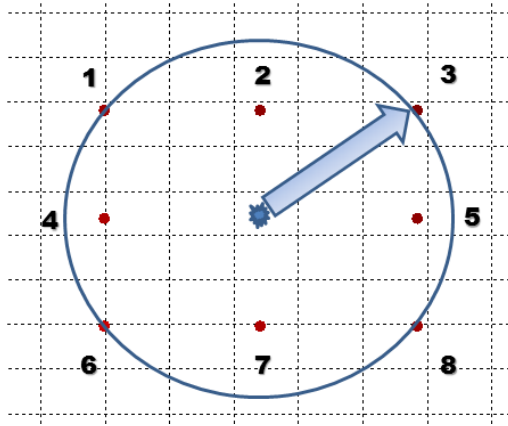


Figure 31 Center of cluster and contributed sensors

Tests for capturing the ability of “center of clusters” interpolation accuracy, and MSE values over surface were run between conventional IDW and our modified version. Table below demonstrates these results.

Table 6 MSE values from IDW and M-IDW

Center of Cluster Index	Center of Cluster Actual Value	IDW	M-IDW	MSE		
					IDW	M-IDW
1	0.7043	0.7002	0.7002	1st run	0.0235	0.0236
2	0.4976	0.4732	0.4732	2nd run	0.0229	0.0229
3	1.003	1	1.001	3rd run	0.0186	0.0191
4	1.107	1.093	1.093	4th run	0.02	0.0206
5	0.162	1.156	1.156	5th run	0.021	0.022
6	1.11	1.1	1.1	6th run	0.0257	0.0267

3.9 Conclusion

The objective of this work is to identify a method suitable for spatially interpolating sparse air temperature observations. The results of this series of experiments indicate that TGF does not have significant advantage in accuracy of resolving underlying field. Although TGF does better under certain sensor network topologies (i.e. clustered sensors), it performs worse overall ($\text{Avg.MSE}_{\text{IDW}}=.02$, $\text{Avg.MSE}_{\text{TGF}}=.04$). Since, in the general case, a favorable sensor network topology cannot be assumed the average MSE values are more representative of the accuracy of the method in this general case. Furthermore, TGF requires several highly-sensitive parameters to be fit, which again renders it less generalizable to broad regions. Finally, the modified M-IDW has slightly better ability to calculate the center of cluster value but overall interpolated surface from conventional IDW has a bit closer value to the actual surface than value of M-IDW ($\text{AvgMSE}_{\text{IDW}}=.0219$, $\text{AvgMSE}_{\text{M-IDW}}=.0224$). Thus, the results of this series of experiments identifies M-IDW as the most promising traditional interpolation method, however, it still does not provide an interpolated surface that accurately captures the spatial structure of the underlying data.

Chapter 4

Remote Sensing and its functionality in retrieving Land Surface Temperature

4.1 Introduction

Variability in meteorological state variables, such as daily maximum, and minimum temperature, that are poorly resolved by regional scale observation result in limitations in simulation of regional land atmosphere interactions. Surface temperature provides wide information on the state of the land-atmosphere interface, and is the driving factor of broad range of fundamental ecological and hydrological phenomena. Temperature data for regional land surface modeling is available from weather stations. Measurement uncertainty and sparse network coverage, lead to inconsistent sampling over the desired region (Jones et al. 2010 [39]). Moreover, the density of meteorological stations is low to capture a spatially distributed characterization since there is no data for some large regions (Marquinez et al. 2003 [40]). The other problem of air temperature interpolation and extrapolation from the ground meteorological stations is that air temperature depends on many local parameters that affect its estimation such as; altitude, sun expose, terrain concavity and distance from coast which cannot be measured by conventional interpolation and extrapolation methods. These limitations encouraged meteorologists to

focus on satellite remote sensing for tasks such as simulation of regional land atmosphere interactions. Satellite sensors provide observations for monitoring, and predicting regional and urban air parameters. These sensors provide comprehensive geospatial information with both qualitative imagery and quantitative data (Engel et al. 2005 [41]).

Constraints on the spatial resolution of early satellites resulted in measurement uncertainty, as Becker and Li 1990 [42], Dozier and Wan 1994 [43]; concluded that regional weather and climate prediction was not possible to perform operationally due to large uncertainties in the retrieval algorithms. They stated that spatial scales of 8 km will cause 1-4 K measurement uncertainties.

One of earliest algorithms for the remote sensing of land surface temperature has been introduced by Brutsaert et al. (1993) [44] which used surface temperatures (LST) derived from measurements by the advanced very high resolution radiometer (AVHRR) on-board the NOAA-9 satellite integrated with collocated temperature and wind soundings in the boundary layer to determine surface sensible and latent heat fluxes from forests in southwestern parts of France. They came up with conclusion that it is necessary to achieve high accuracy in the satellite-retrieved LST in order to reduce the uncertainty in the estimated surface latent heat fluxes. Sun et al. 2002 [45]; retrieved land surface temperature from NOAA/AVHRR data. They applied a split-window technique for atmospheric corrections.

Sun and Pinker (2003; 2009) [46, 47] introduced algorithms for estimating of LST using observations from the Geostationary Operational Environmental Satellite (GOES). The GOES system uses geosynchronous satellites (a geosynchronous satellite is a satellite that

orbits such that coverage repeats regularly over points on the earth over time) which have been a basic element of United States weather monitoring and forecasting since 1974.

One of the easiest methods to estimate air temperature is introduced by Mendez (2004) [48] in the Limpopo river watershed in Mozambique where air temperature is estimated from vertical temperature profiles provided by the moderate resolution imaging spectroradiometer (MODIS) sensors. Later, (Flores et al. 2009 [49]) estimated air temperature over Chile through atmospheric profiles provided by MODIS sensor on a regional level.

In the following section reviews about the United States' Earth Observing System with a special focus on about the sensor used in this research, the moderate resolution imaging spectrophotometer (MODIS).

4.2 Earth Observing System (EOS)

The United States Congress instituted the U.S. Global Change Research Program in 1990 for purpose of understanding the importance of studying global dynamics. NASA's Earth Science Enterprise is making extensive contributions to the program through its Earth Observing System (EOS). Through EOS, National Aeronautics and Space Administration (NASA) is gathering the national and international scientific communities to design, develop, and implement advanced satellite sensors that collect data across a wide spectrum of energy including of ultraviolet (UV), visible, infrared (IR), and microwave. These sensors enable NASA to collect multi-year data sets that will help to answer questions about global change.

4.3 Moderate Resolution Imaging Spectrophotometer (MODIS)

The Moderate Resolution Imaging Spectrophotometer (MODIS) is a multi-spectral imaging instrument that is carried aboard both the Terra and Aqua satellites; it takes measurements all day, every day; and it has a wide field view. Coverage frequency of MODIS system complements other imaging systems such as Landsat's Enhanced Thematic Mapper Plus (ETM+), which covers the earth in higher spatial details (the spatial resolution of ETM+ is 30 m [60 m - thermal, 15-m pan] whereas the spatial resolution of MODIS is 250-1000m), but can only image a given area once every 16 days. Thus, although the ETM+ sensor has a higher spatial resolution, its temporal resolution too low to capture many of rapid biological and meteorological changes that can be observed by the higher temporal resolution MODIS sensor. This resolution trade-off is due to the fact that, spatial and temporal resolutions of a satellite thermal sensor are anti-correlated, meaning that a high spatial resolution is related with low temporal resolution and vice versa (Stathopoulou & Cartalis, 2009 [28]).

The first EOS satellite launched, called Terra, was launched on December 18, 1999, carrying five remote sensors. Terra's³ orbit passes the earth from north to south across the equator in the morning. A second EOS satellite, Aqua⁴ passes south to north over the equator in the afternoon. This tandem coverage, enables the two satellites to capture the

³ Terra is a multi-national, multi-disciplinary mission involving partnerships with the aerospace agencies of Canada and Japan. Managed by NASA's Goddard Space Flight Center, the mission also receives key contributions from the Jet Propulsion Laboratory and Langley Research Center. (Retrieved from <http://modis.gsfc.nasa.gov/>)

⁴ The Aqua mission is a part of the NASA-centered international Earth Observing System (EOS). Aqua was formerly named EOS PM, signifying its afternoon equatorial crossing time. (Retrieved from <http://modis.gsfc.nasa.gov/>)

same area of the earth in the morning and the afternoon, the two satellites will help scientists ensure the accuracy of MODIS and the other instruments aboard Terra and Aqua by optimizing cloud-free remote sensing of the surface and diminishing any optical effects like shadows or brightness that are unique to morning or afternoon sunlight. Terra MODIS and Aqua MODIS viewing the entire Earth's surface every 1 to 2 days, getting data in 36 spectral bands, or groups of wavelengths. These data will improve understanding of global dynamics and processes occurring on the land, in the oceans, and in the lower atmosphere.

MODIS plays an important role in the development of validated, global, interactive Earth system models able to simulate, and predict global changes accurately which assists the policy makers in making vital decisions concerning the environment protection.

4.3.1 MODIS technical Specifications

Detailed MODIS technical Specifications have been listed in the following table.

Table 7 MODIS technical Specifications

Orbit	705 km			
Scan Rate	20.3 rpm, cross track			
Swath Dimensions	2330 km (across track) by 10 km (along track at nadir)			
Size	1.0 x 1.6 x 1.0 m			
Weight	250 kg			
Power	225 W (orbital average)			
Data Rate	11 Mbps (peak daytime)			
Quantization	12 bits			
Spatial Resolution	250 m (band 1-2), 500 , (bands 3-7), 1000m (bands 8-36)			
Primary Use	Band	Bandwidth	Spectral Radiance⁵	Required SNR⁶
Land/Cloud Boundaries	1	620-670	21.8	128
	2	841-876	24.7	201
Land/Cloud Properties	3	459-479	35.3	243
	4	545-565	29	228
	5	1230-1250	5.4	74
	6	1628-1652	7.3	275
	7	2105-2155	1	110
Ocean color/ Phytoplankton/ Biogeochemistry	8	405-420	44.9	880
	9	438-448	41.9	838
	10	483-493	32.1	802
	11	526-536	27.9	754
	12	546-556	21	750
	13	662-672	9.5	910
	14	673-683	8.7	1087
	15	743-753	10.2	586
	16	862-877	6.2	516
Atmospheric Water Vapor	17	890-920	10	167
	18	931-941	3.6	57
	19	915-965	15	250

⁵ (W/m²-μm-sr)

⁶ SNR=Signal-to-noise ratio

Primary Use	Band	Bandwidth	Spectral Radiance	Required NE Δ T ⁷
Surface/Cloud Temperature	20	3.660-3.840	0.45	0.05
	21	3.929-3.989	2.38	2
	22	3.929-3.989	0.67	0.07
	23	4.020-4.080	0.79	0.07
Atmospheric Temperature	24	4.433-4.498	0.17	0.25
	25	4.482-4.549	0.59	0.25
Cirrus Clouds	26	1.360-1.390	6	1503
Water Vapor	27	6.535-6.895	1.16	0.25
	28	7.175-7.475	2.18	0.25
	29	8.400-8.700	9.58	0.05
Ozone	30	9.580-9.880	3.69	0.25
Surface/Cloud Temperature	31	10.780-11.280	9.55	0.05
	32	11.770-12.270	8.94	0.05
Cloud Top Altitude	33	13.185-13.485	4.52	0.25
	34	13.485-13.785	3.76	0.25
	35	13.785-14.085	3.11	0.25
	36	14.085-14.385	2.08	0.35

It should be stated that NASA report Bands 1 to 19 in nanometer scale (nm), while they use micrometer (μ m) scale for Bands 20 to36.

⁷ NE Δ T=Noise-equivalent temperature difference

4.3.2 File Format of land surface temperature (LST) Products

The MODIS land surface temperature (LST) products are produced as a series of seven products which are archived in Hierarchical Data Format – Earth Observing System (HDF-EOS) format files. HDF is the standard archive format for EOS Data information System (EOSDIS) products which is developed by the National Strength and Conditioning Association (NSCA).

The LST product files contain attributes (metadata) and scientific data sets (SDS) with local attributes. Summary of the MODIS LST data products is demonstrated in table 8. A level (L2) product is a geophysical product in latitude and longitude orientations; it has not been temporally and or spatially manipulated, while level 3 (L3) product is a geophysical product which has been manipulated temporally and or spatially [50].

Table 8 MODIS LST data products

Earth Science Data Type (ESDT)	Product Level	Nominal Data Array Dimensions	Spatial Resolution	Temporal Resolution	Map Projection
MOD11_L2	L2	2030 or 2040 lines by 1354 pixels/line	1km at nadir	Swath(scene)	None (lat, lon Referenced)
MOD11A1	L3	1200 rows by 1200 columns	1km actual .927km	Daily	Sinusoidal
MOD11B1	L3	200 rows by 200 columns	6km actual 5.56km	Daily	Sinusoidal
MOD11A2	L3	1200 rows by 1200 columns	1km actual .927km	Eight days	Sinusoidal
MOD11C1	L3	360° by 180° (Global)	0.05° by 0.05°	Daily	Equal-angle geographic
MOD11C2	L3	360° by 180° (Global)	0.05° by 0.05°	Eight days	Equal-angle geographic
MOD11C3	L3	360° by 180° (Global)	0.05° by 0.05°	Monthly	Equal-angle geographic

4.3.3 MOD 11 - Land surface temperature (LST) and Emissivity

Aqua MOD11 and Terra MOD11 products are similar to each other. They both contain Level 2 and 3 LST and emissivity retrieved from MODIS data at spatial resolutions of 1 km and 5 km over global land surfaces under cloudless conditions. The generalized split-window LST algorithm will be applied to retrieve LST for MODIS pixels with known emissivity in bands 31 and 32 (mid infrared range). The physics based day/night LST algorithm is used to simultaneously retrieve surface band emissivity and temperatures from a pair of daytime and nighttime MODIS observations in bands 20, 22, 23, 29, and 31-33 over all types of land cover.

The Terra and Aqua and Aqua platforms provide the daily level 3 Land Surface Temperature at 1km spatial resolution in a tile raster format gridded in the Sinusoidal projection. These products are called MOD11A1 and MYD11A1, respectively. Level 3 products are geophysical products that have been temporally and/or spatially manipulated, and are usually in a gridded map projection format referred to as tiles which have a spatial extent of 10 degrees latitude by 10 degrees longitude at the equator. The tile coordinate system starts at point (0,0) in the upper left corner and proceeds right (horizontal) and downward (vertical). The first element is the horizontal tile number, and the second element corresponds to the vertical tile number.

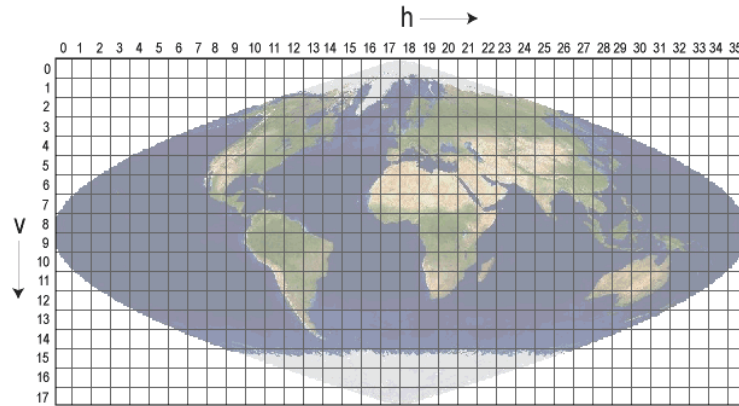


Figure 32 Gridded tile map projection format

The state of New Jersey has been located within boundary of two tiles. These tiles include of (h12, v04), and (h12, v05). Figure below captured Eastern part of United States included gridded tiles, Yellow oval has pointed New Jersey area.

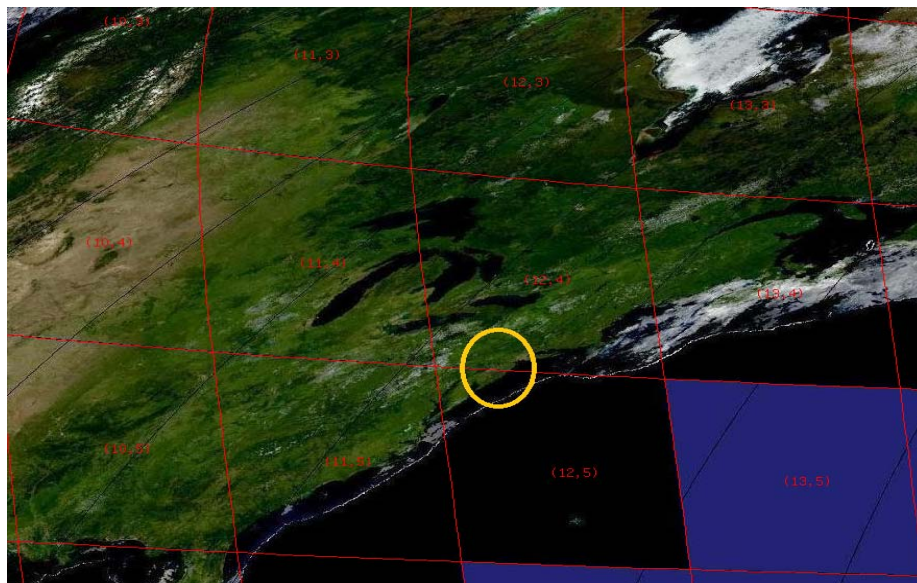


Figure 33 New Jersey location with respect to tile projection

4.4 Image Processing

4.4.1 Converting HDF formatted files to GeoTIFF images

Working with HDF formatted files is not convenient within the MATLAB mapping toolbox environment. Moreover, in order to integrate the data from multiple sensors, a common coordinate system must be applied, a process frequently referred to as data registration. Thus, reprojection of the raw MOD11A1 and MYD11A1 data from the sinusoidal projection to NJ state plane coordinates was performed. NJ State Plane coordinates (NJSPC) are in US Survey Feet units, and are referenced to the North American Datum of 1983 (NAD83) horizontal geodetic datum. Appropriate range of NJSPC values for Easting is approximately between 193000 and 660000 ft. while appropriate range for Northing is approximately between 35000 and 920000ft.

GDAL (Geospatial Data Abstraction Library) software was applied to reproject our HDF files and format the reprojected data in Geographical Tagged Image File Format (GeoTIFF) image files. GeoTIFF is an open source metadata standard which allows georeferencing information to be fixed within a Tagged Image File Format (TIFF) file. The potential additional information includes map projection, coordinate systems, ellipsoids, datums, and everything else necessary to create the exact spatial reference for the file. The sample of MOD11A1 Terra GeoTIFF images for 08/05 2006 are demonstrated as below which include LST profile over the region:

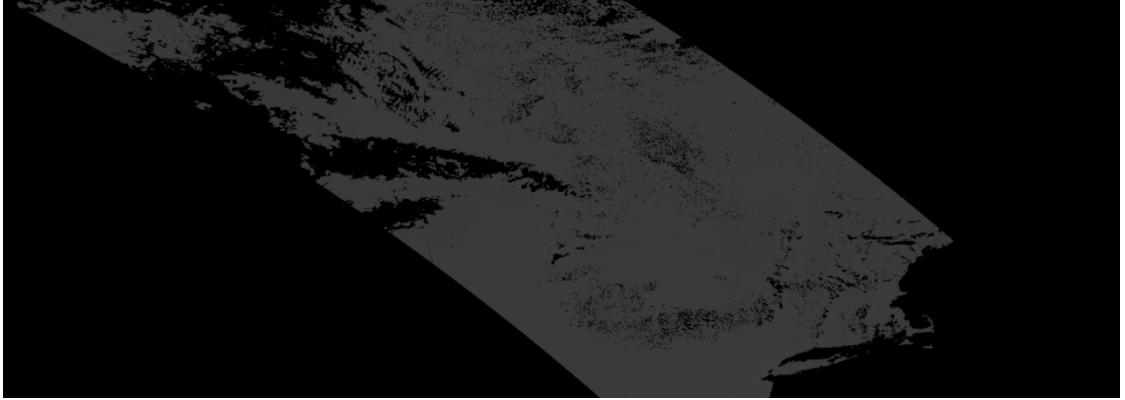


Figure 34 Terra h12v04, 08/05/2006



Figure 35 Terra h12v05, 08/05/2006

4.4.2 MODIS outliers

The MODIS11A1 data, has a nominal spatial resolution of 1km (actual resolution: 0.927 km) which means each pixel size is 0.927 km by 0.927 km. logically the difference between temperature values of neighbor pixels, specifically for the regions which has same land use, should be in a same range in each direction (North to South, and West to East).

Exploratory analysis on the case study region revealed that there were many anomalous pixels that exhibited temperatures that were dramatically higher or lower than their

immediate neighbors. For instance, on 06/13/2006 a 3 by 3 set of pixels near the Burlington county border had an average temperature of 82°F whereas a single pixel in the center of this grouping had a value of 102°F. For the purpose of this study, such pixels were considered as MODIS outliers. Within the MODIS data, the difference between every adjacent pixels in main directions (North to South, and East to West) was calculated. The probability density plot of these residuals are provided as below where EWdata corresponds to pixel by pixel temperature residuals in East to West direction, and NSdata corresponds to pixel by pixel temperature residuals in North to South direction.

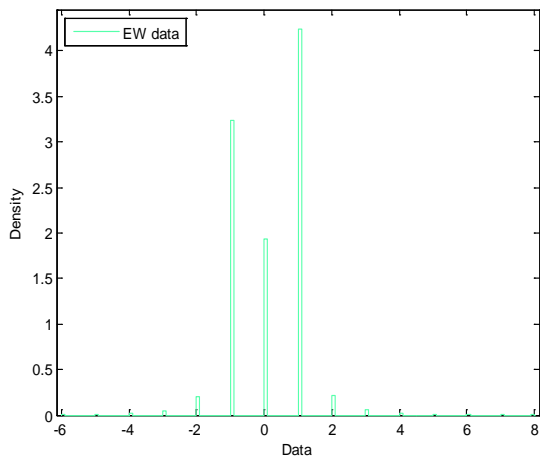


Figure 36 EWdata density plot

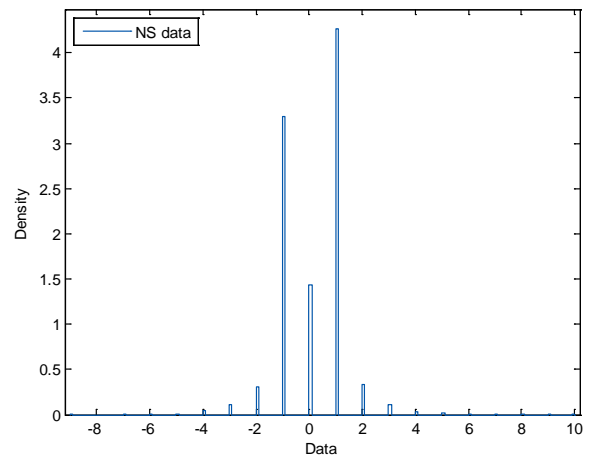


figure 37 NSdata density plot

Probability plots of same data:

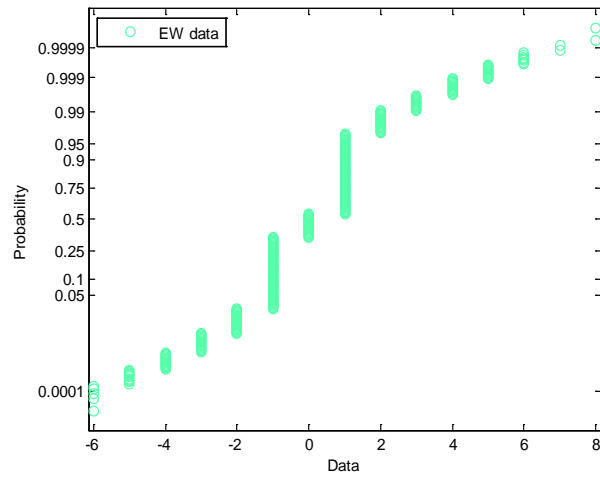


Figure 38 EWdata probability plot

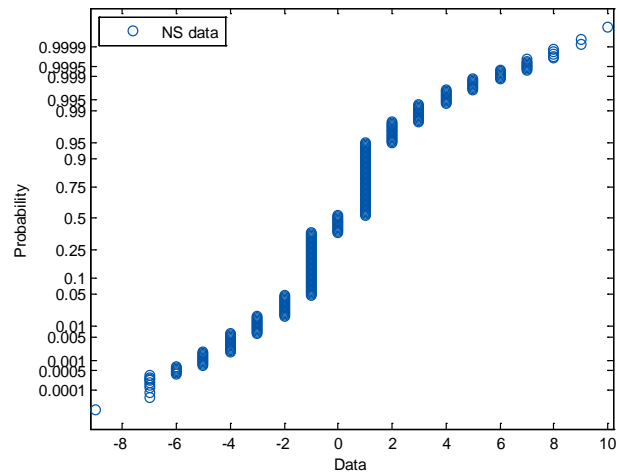


Figure 39 NSdata probability plot

Using these distributions, a threshold filter was defined to remove outliers. This filter as defined to remove pixels that have an absolute difference of 3.2906σ (σ : standard deviation), which represents a 99.9% confidence interval around the local average. Thus, we can expect that the filter will only misclassify valid measurements 0.1% of the time.

4.5 Ground-Based Temperature (TG) Vs. MODIS-Based Temperature (TM)

The following figures illustrate the representativeness of the MODIS-based temperature (TM) for capturing the behavior of the daily min/max temperature measured by the Cooperative Observer Network Gauges. Data from August 2006 over our case study region was selected for this qualitative analysis. Following graphs demonstrate daily Ground-Based Maximum and Minimum Temperature for meteorological stations which had available data over selected time period (23 stations) in addition of corresponded Terra and Aqua delivered temperature. The values which lay directly on the x-axis represent the case where no data available from either the MODIS or Cooperative Observer Network sensors.

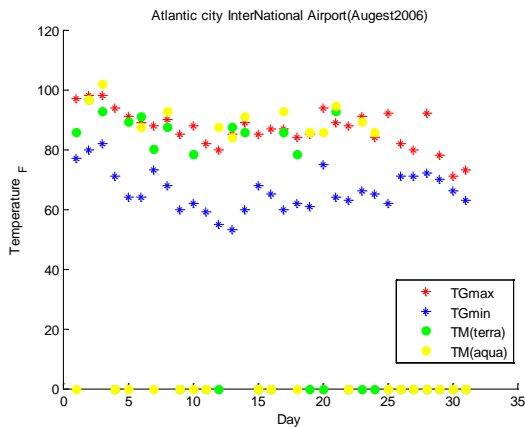


Figure 40 Atlantic city International Airport TG versus TM

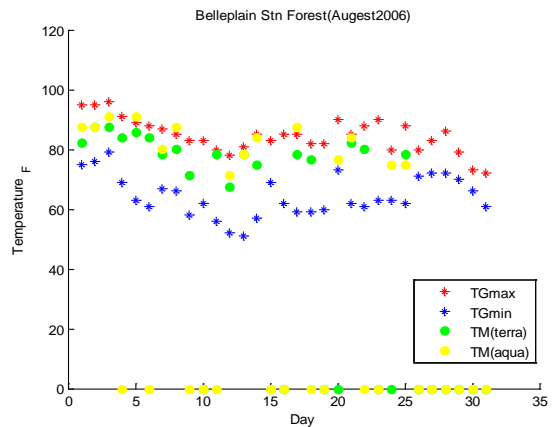


Figure 41 Belleplain Stn Forest TG versus TM

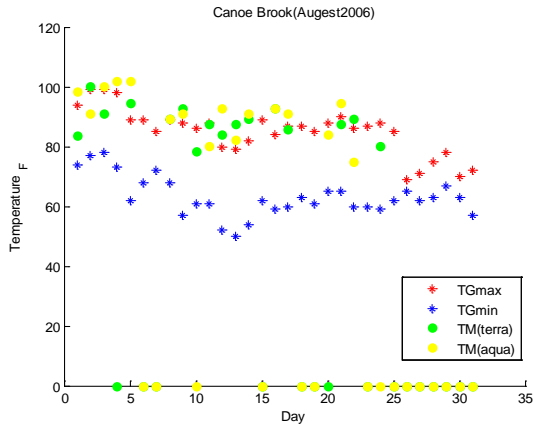


Figure 42 Canoe Brook TG versus TM

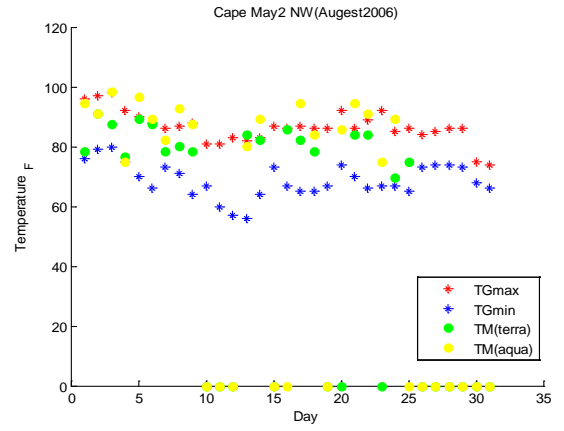


Figure 43 Cape May 2 NW TG versus TM

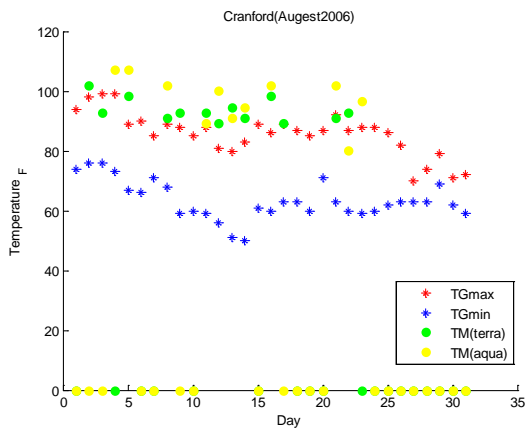


Figure 44 Cranford TG versus TM

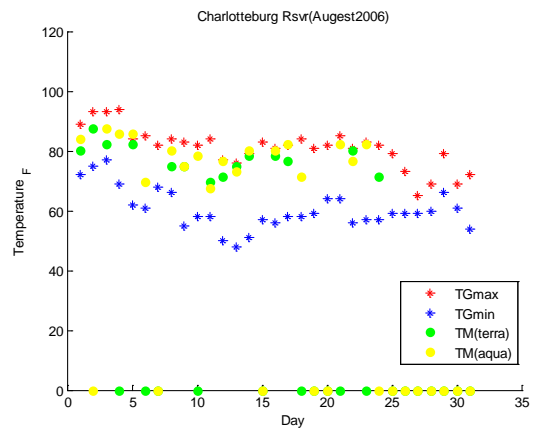


Figure 45 Charlottesburg TG versus TM

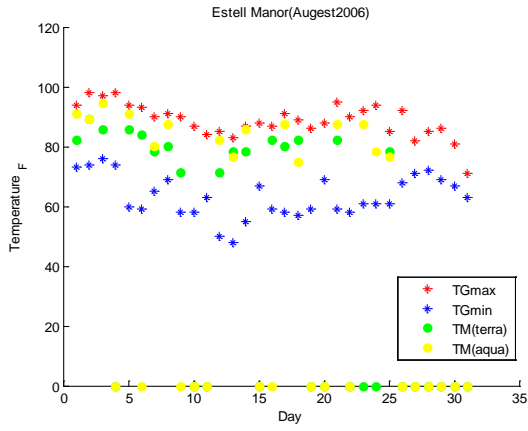


Figure 46 Estell Manor TG versus TM

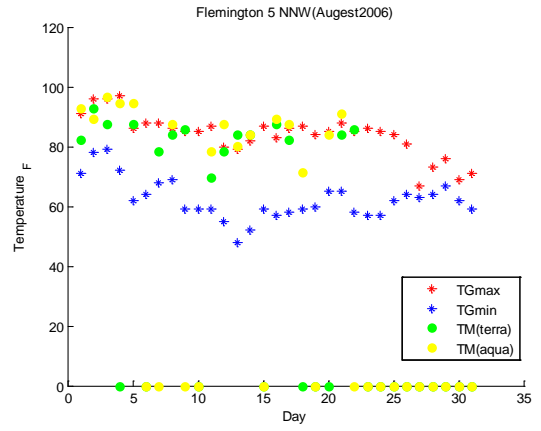


Figure 47 Flemington 5 NNW Tg versus TM

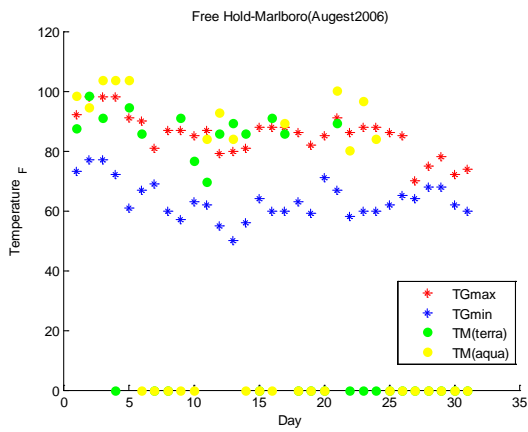


Figure 48 Free Hold-Marlboro TG versus TM

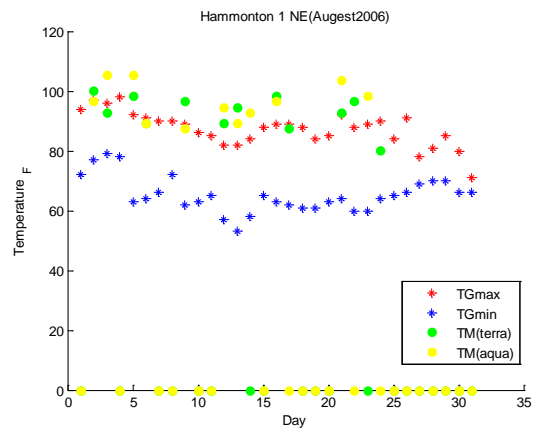


Figure 49 Hammonton 1 NE TG versus TM

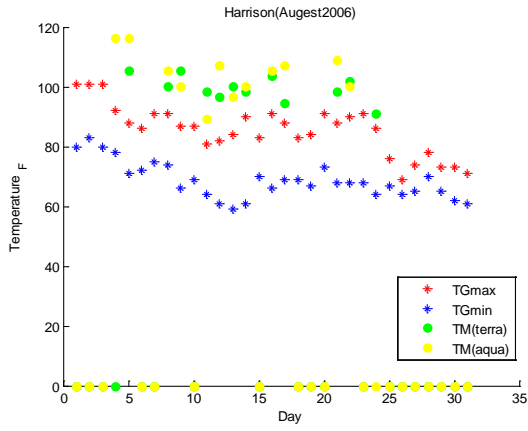


Figure 50 Harrison TG versus TM

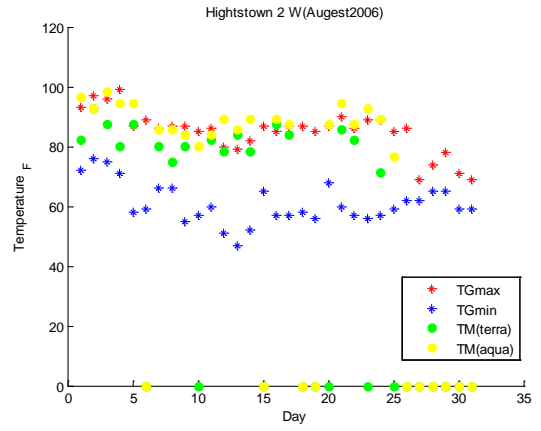


Figure 51 Hightstown 2 W TG versus TM

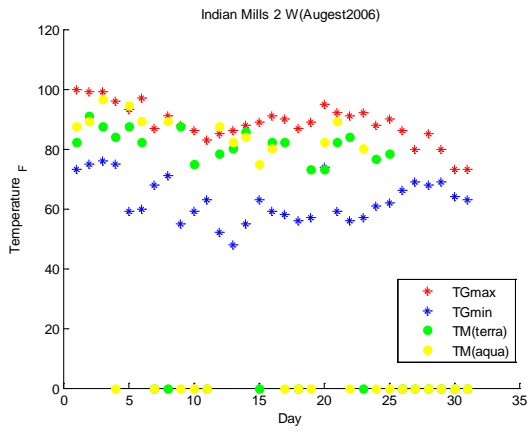


Figure 52 Indian Mills 2 W TG versus TM

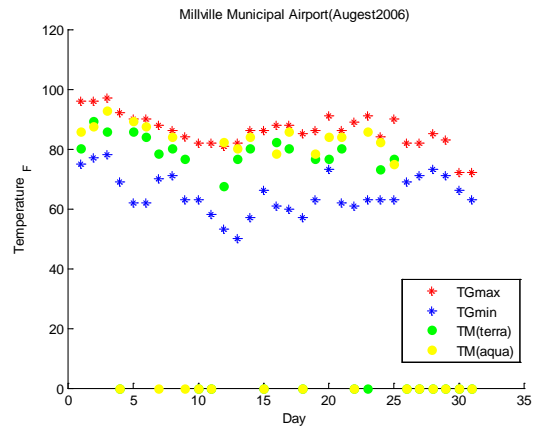


Figure 53 Millville Municipal Airport TG versus TM

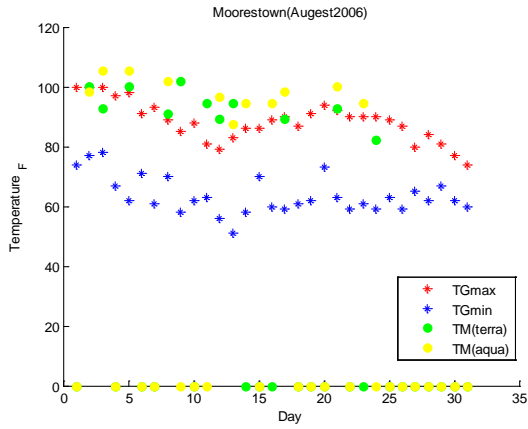


Figure 54 Moorestown TG versus TM

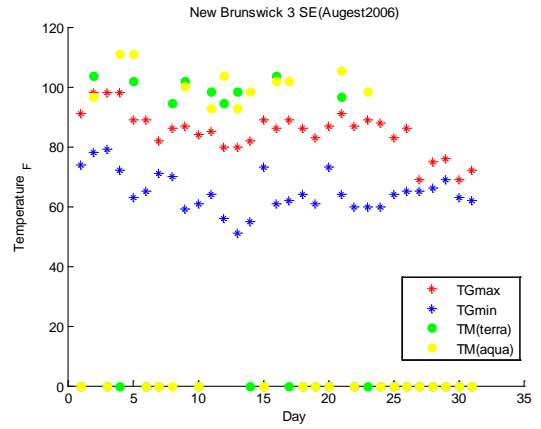


Figure 55 New Brunswick 3 SE TG versus TM

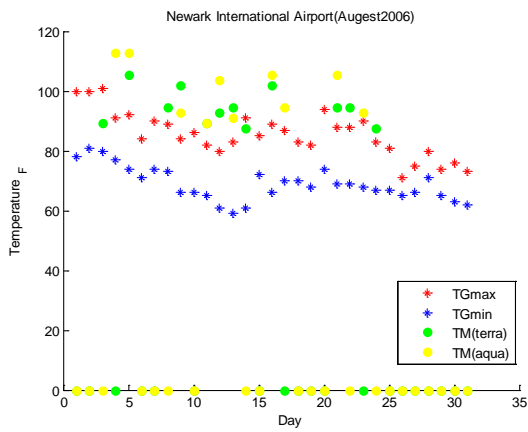


Figure 56 Newark International Airport TG versus TM

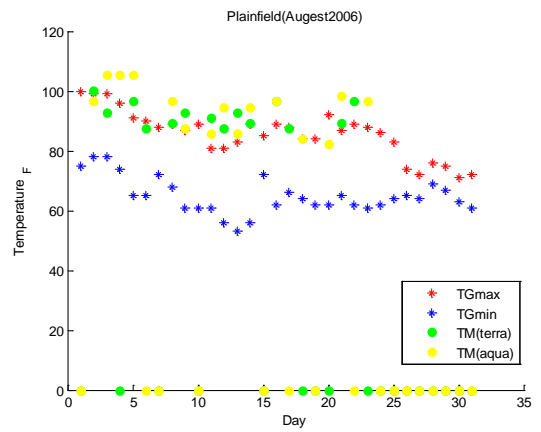


Figure 57 Plainfield TG versus TM

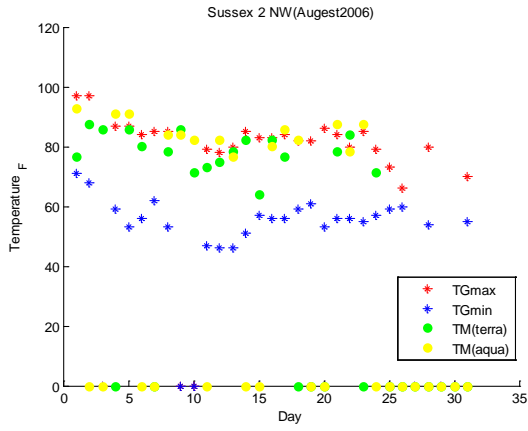


Figure 58 Sussex 2 NW TG versus TM

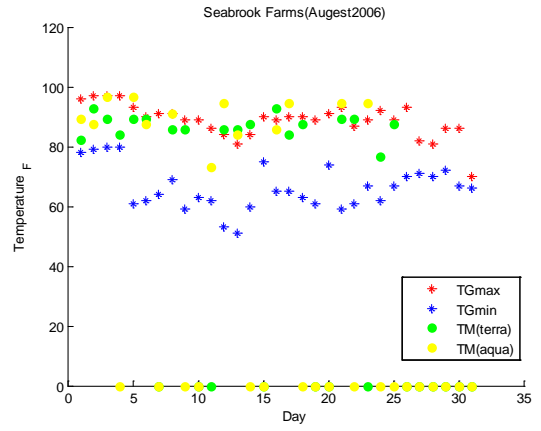


Figure 59 Seabrook Farms TG versus TM

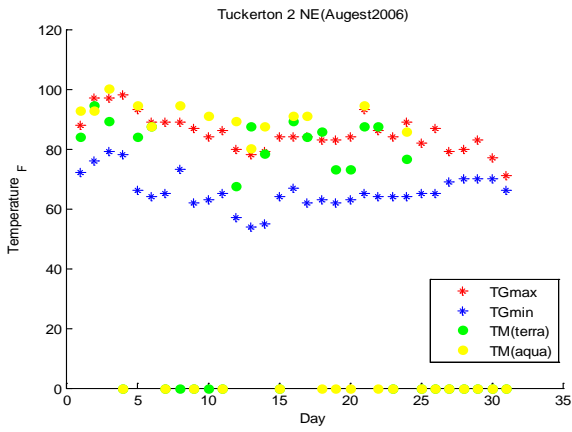


Figure 60 Tuckerton TG versus TM

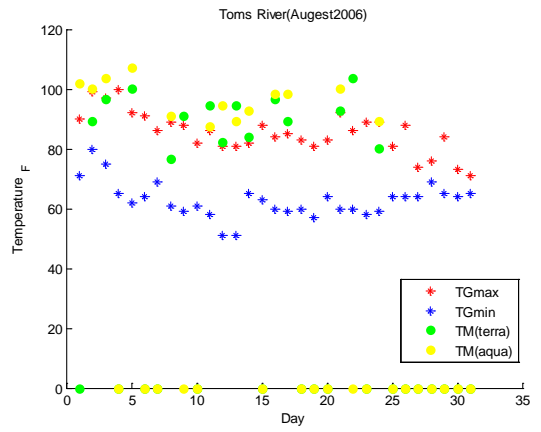


Figure 61 Toms River TG versus TM

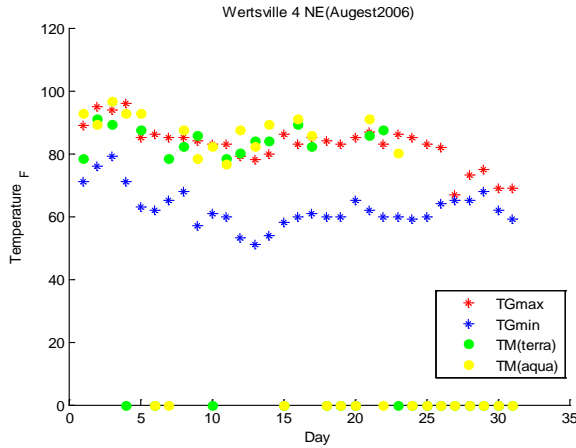


Figure 62 Wertsville 4 NE TG versus TM

Station graphs show that TM (terra) & TM (aqua) are more likely to appear near the TG (max) trend than TG (min) trend. Which indicate us the potential of existent correlation between TG (max) & TMs is higher than that of between TG (min) & TMs. Moreover, it can be illustrated that generally TG (max) and TM (terra) are more alike than other pairs of temperature data. This will be quantified and concluded in the following sections.

4.6 Mathematical Integration of Ground-based & Remote-sensed Temperature Data

A linear relationship between the ground-based temperature and remote-sensed temperature can be defined as:

$$T_G = f(T_M) \tag{16}$$

$$T_G = a.T_M + b = f(T_M) \tag{17}$$

The difference between T_G and T_M represents the error distribution of the linear model:

$$T_G - T_M = \varepsilon_{error\ distribution} \quad (18)$$

The error distribution can be defined either locally or regionally, where local error refers to a pairwise comparison at each station location, and regional error refers to the residual distribution of all pairs of stations in the region as shown below:

$$Each\ station\ location: T_G - T_M = \varepsilon_{T_G, T_M}^{local} \quad (19)$$

$$All\ station\ location: T_G - T_M = \varepsilon_{T_G, T_M}^{all\ locations} \quad (20)$$

Determining whether ε_{T_G, T_M} is generalizable (i.e. not location dependent) will reveal whether a spatially independent model can be used to translate the MODIS temperature data to estimate daily min/max temperature, or whether a spatially dependent model is necessary. The following section discusses how spatial dependency was tested.

4.7 Test of spatial dependency

In statistics, spatial dependence is a measure of the associative dependence between independently measured values in a temporally or *in situ* ordered set, determined in samples selected at positions with different coordinates in a sample space. Spatial dependency is verified by applying analysis of variance to the variance of a set and the first variance term of the ordered set (closest/shortest lag) and comparing the observed F-value between these variances with values of F-distributions at 5% probability with applicable degrees of freedom. In MATLAB, `vartest2(X,Y)` performs an *F* test of the hypothesis that two independent samples, in the vectors X and Y, come from normal

distributions with the same variance, against the alternative that they come from normal distributions with different variances. The result is $H = 0$ if the null hypothesis (variances are equal) cannot be rejected at the 5% significance level, or $H = 1$ if the null hypothesis can be rejected at the 5% level. Table below demonstrates the results of spatial dependency test.

Table 9 Results of spatial dependency test

	Stations Indices																						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
April	n/a	0	0	0	0	1	0	0	0	0	n/a	0	0	0	0	0	0	0	0	0	1	0	n/a
May	1	0	0	0	0	1	0	0	0	0	n/a	0	0	0	0	0	0	0	0	0	0	0	0
June	1	0	0	0	0	1	0	1	0	0	n/a	0	0	0	0	0	0	0	0	0	0	0	0
July	1	1	1	n/a	1	1	1	1	1	1	1	1	1	1	1	1	1	n/a	1	1	1	1	1
Aug	0	1	1	0	1	1	1	0	0	0	n/a	0	0	0	0	0	0	0	0	1	1	1	1
Sep	0	1	0	0	0	1	0	0	0	0	n/a	0	0	0	0	0	0	0	1	1	0	0	0

This result can be interpreted as; the general error distribution independent of spatial location does not exist, and this error should be applied locally.

4.8 Error distribution over the state of New Jersey

In order to understand the relationship between the ground-based and remote-sensed temperature residuals ($T_G - T_M$), and ancillary anthropogenic and microclimatologic features over the state of New Jersey, the plots of these residuals for available stations have been provided as below. The background of each map represents digital elevation model, and for each station location the population density value has been attached. The size of white circles corresponds to the magnitude of residual in such a way that, the

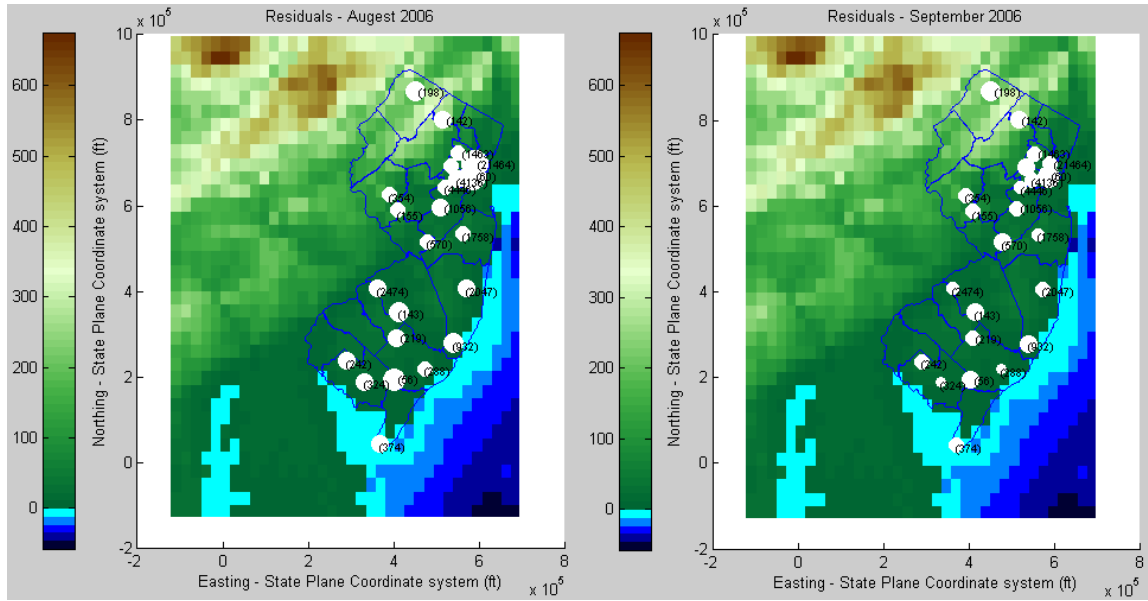


Figure 63 Residual distribution plot (Left: August, Right: September)

New Jersey is the most heavily urbanized state in the US, and it is proximal to the 1st (New York City) and 6th (Philadelphia) largest cities in the US. As it can be seen from the images the biggest magnitudes of residuals are corresponded to the regions close to New York City which possess the highest population density over the state of New Jersey.

4.9 Mathematical function of temperature interpolator model

In order to map the temperature, the following spatial interpolation model was defined:

$$T_i = T_{Mi} + \Delta_S \quad (21)$$

Where, $T_{G,k}$ is the k -th nearest temperature of available meteorological ground-based station, and $T_{M,k}$ is the retrieved temperature of k -th nearest meteorological station from

the MODIS satellite data. The simplest form of this model is for $k=1$, which is comparable with conventional nearest neighbor interpolation algorithm.

if $k = 1$

$$T_i = T_{Mi} + T_{G,k} - T_{M,k} \quad (22)$$

This model is hereafter referred to as the MODIS-enhanced Nearest Neighbor interpolator.

4.10 Leave one out cross validation Test

Leave-one-out cross-validation (**LOOCV**) involves using a single observation from the original sample as the validation data, and the remaining observations as the training data. Here, the validation dataset is ground based and remote sensed temperature residuals of point i ($T_{G,i} - T_{M,i}$), and the training dataset is the same value of nearest neighbor station ($T_{G,k} - T_{M,k}$). The test run with terra data and then with aqua data to capture which one has better correlation with station maximum temperature than the other. Below figures demonstrates test results distribution.

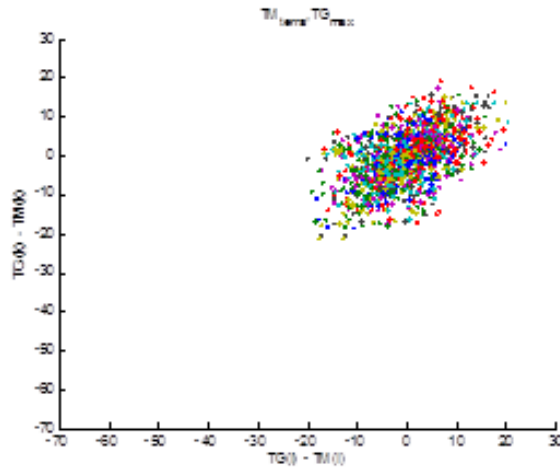


Figure 64 LOOCV plot, Terra versus TGmax

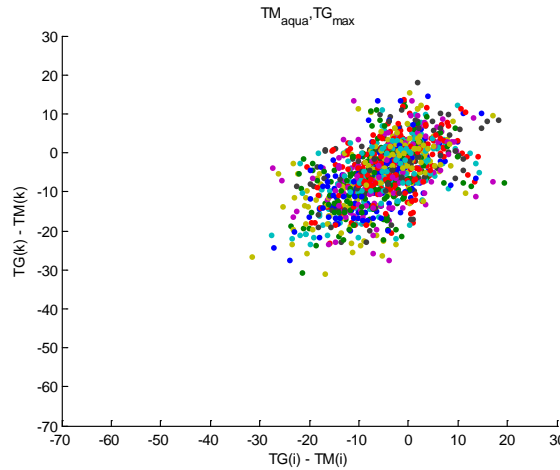


Figure 65 LOOCV plot Aqua versus TGmax

The more squeezed, the more correlation exists. Visually it can be seen that left figure is more squeezed. Moreover, the MSE value of left image is less than MSE value of right one. Which indicates that generally terra data has more correlation with stations maximum temperature than aqua data ($MSE_{Residuals,terra}=6.97$, $MSE_{Residuals,aqua}=7.48$).

4.11 Model Demonstration

In this section, the interpolated temperature map over the state of New Jersey has been demonstrated for two random dates 04/02/2006 and, 06/18/2006. We expect the recent spatial interpolator model quantify anthropogenic, and microclimatological effects better than conventional methods, due to the fact that low density of meteorological stations cannot capture these features. For these two sample dates, Raw MODIS image, conventional nearest neighbor interpolation map, IDW map with 5-NN, and MODIS enhanced nearest neighbor algorithm have been demonstrated. The reason for providing raw MODIS image is to see which model can produce map of temperature with respect to regional parameters such as urban heat effect, proximity to water bodies, altitude, and etc. The first four images correspond to the April second of 2006 in order that, the top left image shows the raw MODIS data, the top right image retrieved from MODIS enhanced interpolation method, the bottom left image is the output of 1-nearest neighbor algorithm, and the bottom right shows the map of inverse distance weighting method. The following next four images correspond to the 18th of June 2006 in the same line up.

Maps of temperature for 04/02/2006

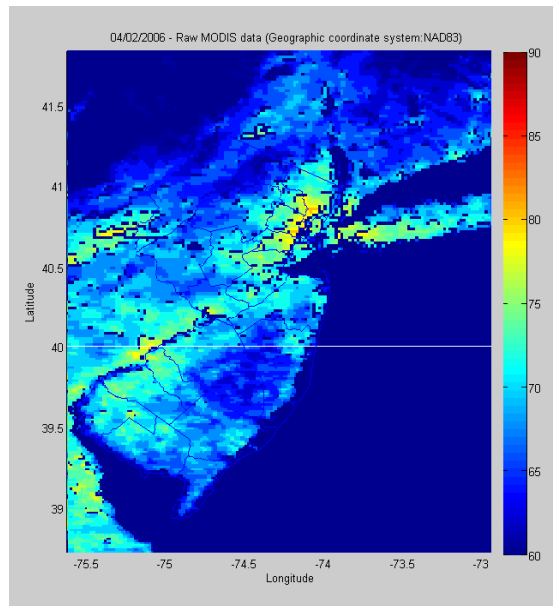


Figure 66 Raw MODIS data 04/02/2006

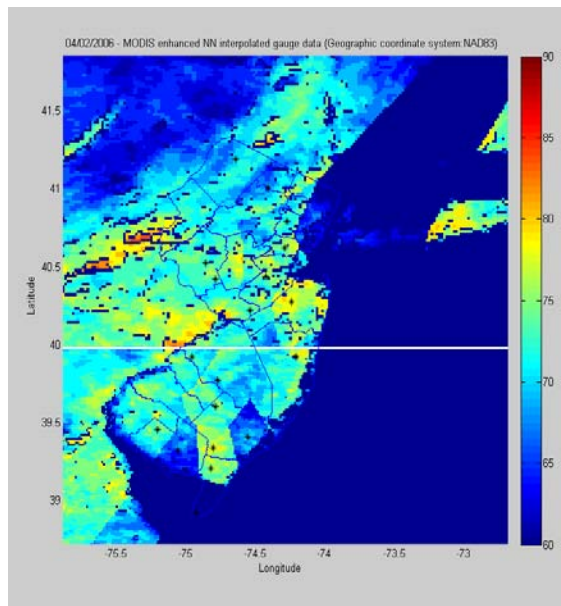


Figure 67 MODIS enhanced NN interpolated gauge data 04/02/2006

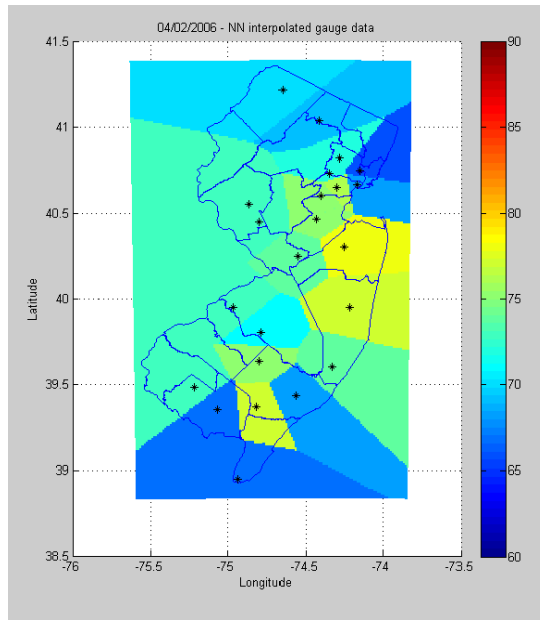


Figure 68 NN interpolated gauge data 04/02/2006

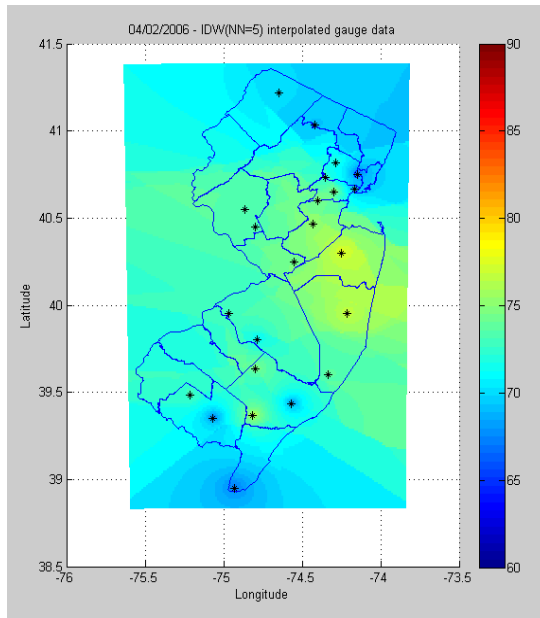


Figure 69 IDW interpolated gauge data 04/02/2006

Maps of temperature for 06/18/2006

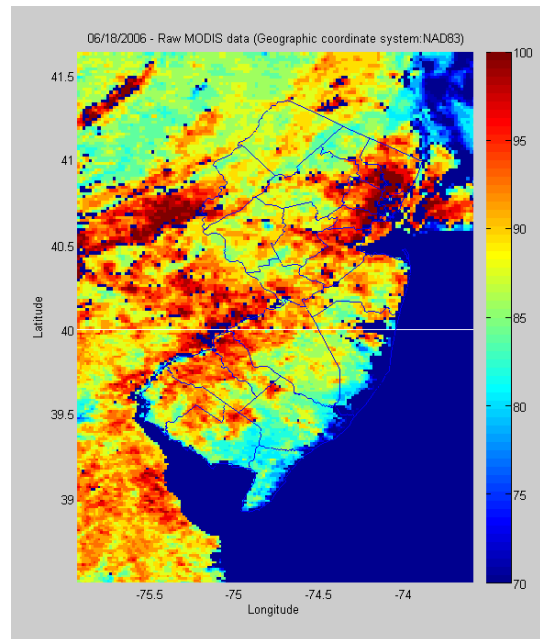


Figure 70 Raw MODIS data 06/18/2006

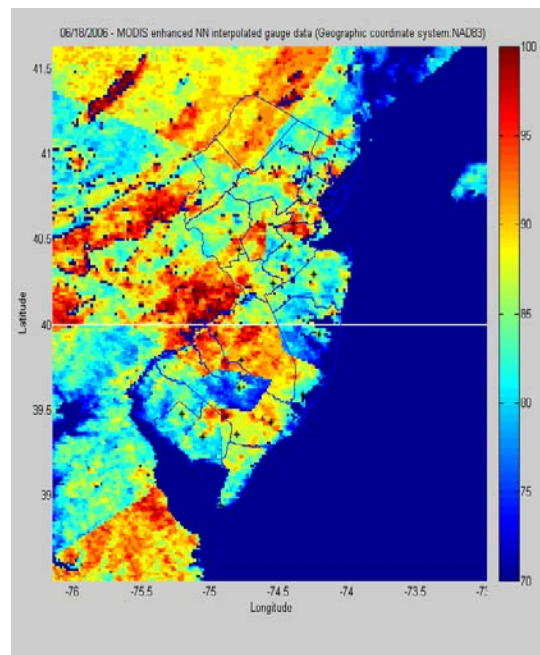


Figure 71 MODIS enhanced NN interpolated gauge data 06/18/2006

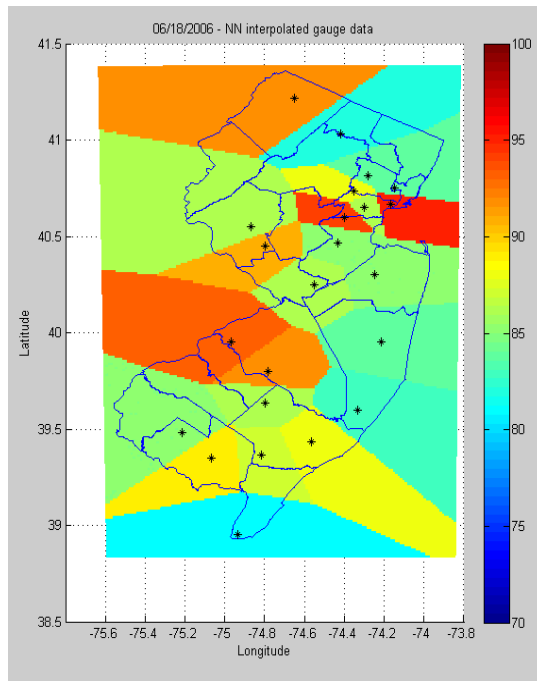


Figure 72 NN interpolated gauge data 06/18/2006

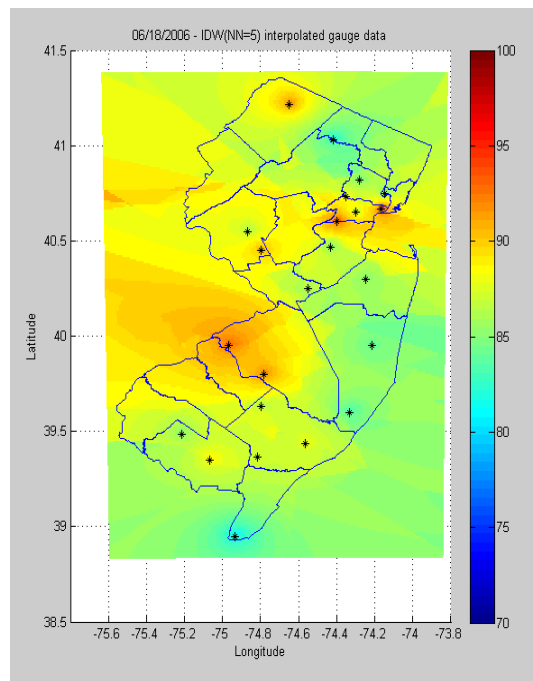


Figure 73 IDW interpolated gauge data 06/18/2006

4.12 Conclusion and results

The most notable achievement of new introduced method as it can be seen from the interpolated maps is the ability of the model to capture temperature pattern revealed by the remote sensing observations. This is the advantage of integrating both remote-sensed and ground-based data with each other. This integration allows us to explore the relationship between climatic elements (in our case, air temperature) as the dependent variable and climatic factors (independent geographic factors) better than conventional methods. Comparison of the mean squared error (MSE) values indicate that the simple format of the MODIS enhanced nearest neighbor method, which uses one nearest neighbor to determine the spatially dependent representativeness error between the MODIS sensor and the ground based min/max temperature, is more accurate than the conventional nearest neighbor method ($MSE_{\text{Conventional 1-NN}}=0.84$. $MSE_{\text{MODIS enhanced 1-NN}}=0.81$). Although the MSE value of IDW ($MSE_{\text{IDW NN=5}}=0.51$) is slightly less than the MSE value of MODIS enhanced method, the IDW method over smooths the temperature map. Thus, the new method produces a result with a similar estimation accuracy of IDW, but with a better characterization of the spatial variation of min/max temperature.

Chapter 5

Conclusion and Discussion

Spatial interpolation of sparse air temperature observations has been a challenge among scientists which has arisen several spatial interpolation methods. In this study two fundamental weighted average models has been compared. Series of experiments indicate that TGF does not have significant advantage in accuracy of resolving underlying field. Although TGF does better under certain sensor network topologies (i.e. clustered sensors), it performs worse overall ($Avg_{MSE\ IDW}=.02$, $Avg_{MSE\ TGF}=.04$). Since, in the general case, a favorable sensor network topology cannot be assumed the average MSE values are more representative of the accuracy of the method in this general case. Furthermore, TGF requires several highly-sensitive parameters to be fit, which again renders it less generalizable to broad regions. The modified M-IDW has slightly better ability to calculate the center of cluster value but overall interpolated surface from conventional IDW has a bit closer value to the actual surface than value of M-IDW ($Avg_{MSE,IDW}=.0219$, $Avg_{MSE,M-IDW}=.0224$). Thus, the results of this series of experiments identifies M-IDW as the most promising traditional interpolation method, however, it still does not provide an interpolated surface that accurately captures the spatial structure of the underlying data. This could be the result of insufficient embedded

temperature data density and complex interactions of the temperature data which makes the characterization of the relationship between local temperature and ancillary anthropogenic and microclimatologic features unclear.

The new approached MODIS enhanced nearest neighbor method allows us to explore the relationship between climatic elements (in our case, air temperature) as the dependent variable and climatic factors (independent geographic factors) better than conventional methods which is the advantage of integrating both remote-sensed and ground-based data with each other. The comparison of the mean squared error (MSE) values indicate that the simple format of the MODIS enhanced nearest neighbor method, which uses one nearest neighbor to determine the spatially dependent representativeness error between the MODIS sensor and the ground based min/max temperature, is more accurate than the conventional nearest neighbor method ($MSE_{\text{Conventional 1-NN}}=0.84$. $MSE_{\text{MODIS enhanced 1-NN}}=0.81$). Although the MSE value of IDW ($MSE_{\text{IDW NN=5}}=0.51$) is slightly less than the MSE value of MODIS enhanced method, the IDW method over smooths the temperature map. Thus, the new method produces a result with a similar estimation accuracy of IDW, but with a better characterization of the spatial variation of min/max temperature.

5.1 Future Work

Our work has established the foundation for daily temperature mapping based on integration of embedded and remote sensed temperature data. Although, this work has reduced the existed gap in the challenge of capturing spatial structure of the underlying temperature map but this work can be expanded by applying and testing other reliable interpolation algorithms on embedded and remote sensed integrated dataset in order to increase the accuracy. More study over potential variety of trends between embedded and remote sensed temperature data could pave the way of reaching to more precise daily interpolated temperature map.

I Appendix

MATLAB Codes:

Map of New Jersey in state plane coordinate system:

```
clc
close all
clear all
S = shaperead('nj_counties.shp')
nx = 100;
ny = nx;
p=5;    %--- #of neighbors for interpolation
scalefactor = 1; latlim = [38.5 41.5]; lonlim = [-76.7 -73.8];
[Z,refvec] = tbase(scalefactor,latlim,lonlim);
mstruct = defaultm('tranmerc');
mstruct.geoid = almanac('earth','grs80','survey foot');
mstruct.origin=[38.8333 -74.5 0];
mstruct.falsenorthing = 0.0;
mstruct.falseeasting = 492125.0;
mstruct.scalefactor = 0.9999;
[lat lon] = minvtran(mstruct, [S.X]', [S.Y]');

% -----Building reference matrix for elevation,based on 'tbase' info
[x11 y11] = mfwdtran(mstruct, 41.5, -76.7); %--northing-easting
pixel(1,1)
[xnn y] = mfwdtran(mstruct, 41.5, -73.8);
[x ynn] = mfwdtran(mstruct, 38.5, -76.7); %--northing-easting
pixel(1,1)
dx = (xnn-x11)/(numel(Z(:,1))-1);
dy = (ynn-y11)/(numel(Z(:,1))-1);
R = makerefmat(x11, y11, dx, dy);
%mstruct = defaultm(mstruct);

% Use projection to convert from state plane to geocoordinates
[lat lon] = minvtran(mstruct, [S.X]', [S.Y]');
figure
plot(lon, lat);

%-----Sensors with complete data-----
----
b= load('availablestatoins112007.txt')
SensorCoords = [ b(:,1) b(:,2)]
[Sensors(1:3).Geometry] = deal('Point');
for i=1 : size(b(:,1));
    Sensors(i).Lat = SensorCoords(i,1); Sensors(i).Lon =
SensorCoords(i,2);
end

%-----Malfunctioned Sensors-----
----
f= load('stationcoordinatesmalfunctioned.txt')
```

```

SensorCoordsMal = [ f(:,1) f(:,2)]
[Sensors(1:4).Geometry] = deal('Point');
for i=1 : 26;
    SensorsMal(i).Lat = SensorCoordsMal(i,1); SensorsMal(i).Lon =
SensorCoordsMal(i,2);
end

% Inspect completed geostruct and its first member
Sensors;
Sensors(1)
figure
plot(lon, lat);
hold on
geoshow(Sensors);
plot(SensorCoords(:,2), SensorCoords(:,1), 'ko');
hold off

% Use projection to convert from geocoords to state plane coords(all
% sensors)
[x y] = mfwdtran(mstruct, SensorCoords(:,1), SensorCoords(:,2));
%Malfunctioned sensors
[xm ym] = mfwdtran(mstruct, SensorCoordsMal(:,1),
SensorCoordsMal(:,2));
% Now plot the sensor locations on the state plane map
figure;
mapshow(S);
hold on
colorbar;
mapshow([S.X]', [S.Y]');
% mapshow(P);
hold on
plot(x,y, 'k*');
hold on
plot(xm,ym, 'y*');

```

Seasonality Removal:

```
clc
clear
close all
%---35 Sensors, "each day of yr" temp represents average of 10yrs data(
if it was
%available)
%--- column1:Tmax , column2: Tmin
%--- h: sensor elevation
%Station letter=SL
%---Curve fitting General format: SinSL = SL0 +
SL1.COS(omega*t)+SL2.SIN(omega*t)

a=load('Athlant.Int.txt');
ha=60*ones(365,1);
figure(1)
scatter([1:size(a(:,1))],a(:,1),'bo')
t=load('t.txt');
omega=2*pi*1/365;
uv=load('365daysOverNJ.txt');
a0max=sum(uv(:,1))/365;
almax=(2/365)*sum(uv(:,1).*cos(omega*t(:)));
a2max=(2/365)*sum(uv(:,1).*sin(omega*t(:)));
Sinamax=a0max+almax*cos(omega*t(:,1))+a2max*sin(omega*t(:,1));
detrendamax=a(:,1)-Sinamax(:,1);
a0min=sum(a(:,2))/365;
almin=(2/365)*sum(uv(:,2).*cos(omega*t(:)));
a2min=(2/365)*sum(uv(:,2).*sin(omega*t(:)));
Sinamin=a0min+almin*cos(omega*t(:,1))+a2min*sin(omega*t(:,1));
detrendamin=a(:,2)-Sinamin(:,1);
hold on
plot(t(:,1),Sinamax,'k')
figure(2)
scatter(t,detrendamax)
figure(3)
scatter([1:size(a(:,1))],detrendamax(:,1),'bo')
```

Autocorrelation and Partial Autocorrelation Functions:

```
clc
clear
close all
%Perform an analysis of the time series to determine stationarity and
seasonality
%-----
----
a1=load('detrendedaclsinosu.txt');
a2=load('vcxz.txt');
data=zeros(3650,2);

data(:,1)=a1;
data(:,2)=a2(1:3650,2);
% data=load('vcxz.txt');
figure(1);
plot(data(:,2), data(:,1));
title('Temperature (F)');
xlabel('Year');
ylabel('Maximum Temperature (F)');
%Calculate linear trend
A=[data(:,2) ones(size(data(:,2)))];
m=(inv(A'*A)*A)*data(:,1);
% Observing 'm' value shows that we dont have linear trend.
% m = 0.0027, which shows slope is significantly zero.
%-----
NLags=700;
NData=size(data,1);
Lags = linspace(0,NLags,NLags+1);
laggedData=zeros(NData-NLags,NLags+1);
for i=1:NData-NLags
    for j=0:NLags
        laggedData(i,j+1)=data(i+j,1);
    end
end
%Calculate **ACF=covariance(X(t),X(t-lag))/variance(X)**
Covariance=cov(laggedData,1);
ACF=Covariance(:,1)/Covariance(1,1);
tmp=cumsum(ACF.^2);
ACFBounds=2*(1/sqrt(NData)*(1+2.*tmp));
figure(2);
plot(Lags,ACF,'ko-')
ylim([-2 2])
title('Auto Correlation Function');
xlabel('Lag')
ylabel('ACF')
%Calculate partial autocorrelation function
PACF=zeros(NLags+1,1);
PACF(1)=1;
for i=1:NLags
    %Yule-Walker Equations
    r=zeros(i,1);
    R=zeros(i,i);
    for j=1:i
        r(j)=ACF(j+1);
```

```

        for k=1:i
            R(j,k)=ACF( abs(j-k)+1 );
        end
    end
    %R
    %r
    PHI=inv(R)*r;
    PACF(i+1)=PHI(i);
end
PACFBounds = ones(NLags+1,1);
PACFBounds = PACFBounds*2*1/sqrt(NData);
figure(3);
plot(Lags,PACF,'ko-')
% ,Lags,PACFBounds,'m-',Lags,-PACFBounds,'m-');
title('PACF of Data');
xlabel('lag');
ylabel('PACF');
legend('PACF','95% Confidence Bounds');

```


Semivariogram Analysis:

```
function [G] = semivariogram( Data, Lags, Dh, Theta, DTheta )
    G = zeros(size(Lags,1),1);
    for i=1:size(Lags,1);
        sum=0;
        N=0;
        for j=1:size(Data,1)
            tmp = [Data(1:j-1,:);Data(j+1:size(Data,1),:)];
            [flag, heads]=findLaggedData(Data(j,:), tmp, Lags(i), Dh,
Theta, DTheta);
            if( flag )
                for k=1:size(heads,1)
                    sum = sum+((Data(j,3)-heads(k,3) )^2);
                    N = N+1;
                end
            end
        end
        G(i)= 1/(2*N)*sum;
    end
end
```

```
Lags = 10000*[5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80]';
for i=1 : 14;
    for j=2:15;
        L(i,j)=sqrt((x(i+1)-x(i))^2+(y(j)-y(j-1))^2)
    end
end
[G]=semivariogram(Data, Lags, 50000, 22.5, 22.5);
```

```
figure(1)
plot(Lags,G,'ko')
xlabel('Distance (ft)');
title('Semivariogram of Data');
```

Grid construction with random placement of sensors:

```
clc
clear
close all

%-----
%-----Grid Construction-----
%-----

%-----Specification of the grid-----
dx = 0.01; %----x spacing
dy = 0.01; %----y spacing
imax = 2; %----max # of points in x direction
jmax = 2; %----max # of points in y direction
%-----# of total cells = (imax-1)*(jmax-1)

%-----Construction-----

for i = 1:imax
    for j = 1 : jmax
        x(i,j) = (i-1)*dx;
        y(i,j) = (j-1)*dy;
    end
end

%-----
%-----Random placement of sensors-----
%-----

N = imax*jmax;
M = 1; %-----total number of sensors
s(1) = unidrnd(N); %-----location of first sensor
i = 2;
while i<=M
    R(i) = unidrnd(N);
    while (min(abs(s(1:i-1)-R(i)))==0)
        R(i) = unidrnd(N);
    end
    s(i) = R(i);
    i = i + 1;
end

%-----
%-----Random temperature assignment-----
%-----

temp=zeros(imax,jmax);

for i = 1:M
    temp(s(i)) = unifrnd(-1,1);
end
```

Inverse Distance Weighting (IDW):

```
clc
close all
clear
p = 5; %-----number of nearest neighbors
[Xtrue,Ytrue]=meshgrid(-3:0.1:3,-3:0.1:3);
a1=1;x01=-3;y01=-3;s1=1.2;v1=1.2;
a2=1;x02=3;y02=3;s2=1;v2=1;
a3=1;x03=0;y03=0;s3=2;v3=2;
%
a4=1;x04=3;y04=-3;s4=1.3;v4=1.3;
z1 = a1*exp(-((Xtrue-x01).^2/2/s1^2)-((Ytrue-y01).^2/2/v1^2));
z2 = a2*exp(-((Xtrue-x02).^2/2/s2^2)-((Ytrue-y02).^2/2/v2^2));
z3 = a3*exp(-((Xtrue-x03).^2/2/s3^2)-((Ytrue-y03).^2/2/v3^2));
z4 = a4*exp(-((Xtrue-x04).^2/2/s4^2)-((Ytrue-y04).^2/2/v4^2));
Ztrue=z1+z2+z3+z4;
figure(1)
surf(Xtrue,Ytrue,Ztrue)
view(0,90)
Nrows = size(Xtrue(:,1),1);
Ncols = Nrows;
Nsamp = 52;
Xsensor =
[Xtrue(2,56),Xtrue(2,58),Xtrue(2,60),Xtrue(4,56),Xtrue(4,60),Xtrue(6,56)
),Xtrue(6,58),Xtrue(6,60),Xtrue(56,2),Xtrue(56,4),Xtrue(56,6),Xtrue(58,
2),Xtrue(58,6),Xtrue(60,2),Xtrue(60,4),Xtrue(60,6),Xtrue(2,1),Xtrue(2,3
),Xtrue(2,5),Xtrue(4,1),Xtrue(6,1),Xtrue(6,3),Xtrue(6,5),Xtrue(4,5),Xtr
ue(30,28),Xtrue(32,28),Xtrue(34,28),Xtrue(30,30),Xtrue(34,30),Xtrue(30,
32),Xtrue(32,32),Xtrue(34,32),Xtrue(40,12),Xtrue(40,10),Xtrue(40,8),Xtr
ue(42,12),Xtrue(42,8),Xtrue(44,12),Xtrue(44,10),Xtrue(44,8),Xtrue(50,48
),Xtrue(50,52),Xtrue(48,50),Xtrue(48,48),Xtrue(48,52),Xtrue(52,48),Xtru
e(52,50),Xtrue(52,52),Xtrue(20,10),Xtrue(40,20),Xtrue(30,30),Xtrue(3,40
),Xtrue(30,5),Xtrue(18,50),Xtrue(60,1),Xtrue(8,8),Xtrue(60,60),Xtrue(60
,45),Xtrue(50,30),Xtrue(17,17)];
Ysensor =
[Ytrue(2,56),Ytrue(2,58),Ytrue(2,60),Ytrue(4,56),Ytrue(4,60),Ytrue(6,56)
),Ytrue(6,58),Ytrue(6,60),Ytrue(56,2),Ytrue(56,4),Ytrue(56,6),Ytrue(58,
2),Ytrue(58,6),Ytrue(60,2),Ytrue(60,4),Ytrue(60,6),Ytrue(2,1),Ytrue(2,3
),Ytrue(2,5),Ytrue(4,1),Ytrue(6,1),Ytrue(6,3),Ytrue(6,5),Ytrue(4,5),Ytr
ue(30,28),Ytrue(32,28),Ytrue(34,28),Ytrue(30,30),Ytrue(34,30),Ytrue(30,
32),Ytrue(32,32),Ytrue(34,32),Ytrue(40,12),Ytrue(40,10),Ytrue(40,8),Ytr
ue(42,12),Ytrue(42,8),Ytrue(44,12),Ytrue(44,10),Ytrue(44,8),Ytrue(50,48
),Ytrue(50,52),Ytrue(48,50),Ytrue(48,48),Ytrue(48,52),Ytrue(52,48),Ytru
e(52,50),Ytrue(52,52),Ytrue(20,10),Ytrue(40,20),Ytrue(30,30),Ytrue(3,40
),Ytrue(30,5),Ytrue(18,50),Ytrue(60,1),Ytrue(8,8),Ytrue(60,60),Ytrue(60
,45),Ytrue(50,30),Ytrue(17,17)];
Zsensor =
[Ztrue(2,56),Ztrue(2,58),Ztrue(2,60),Ztrue(4,56),Ztrue(4,60),Ztrue(6,56)
),Ztrue(6,58),Ztrue(6,60),Ztrue(56,2),Ztrue(56,4),Ztrue(56,6),Ztrue(58,
2),Ztrue(58,6),Ztrue(60,2),Ztrue(60,4),Ztrue(60,6),Ztrue(2,1),Ztrue(2,3
),Ztrue(2,5),Ztrue(4,1),Ztrue(6,1),Ztrue(6,3),Ztrue(6,5),Ztrue(4,5),Ztr
ue(30,28),Ztrue(32,28),Ztrue(34,28),Ztrue(30,30),Ztrue(34,30),Ztrue(30,
32),Ztrue(32,32),Ztrue(34,32),Ztrue(40,12),Ztrue(40,10),Ztrue(40,8),Ztr
ue(42,12),Ztrue(42,8),Ztrue(44,12),Ztrue(44,10),Ztrue(44,8),Ztrue(50,48
),Ztrue(50,52),Ztrue(48,50),Ztrue(48,48),Ztrue(48,52),Ztrue(52,48),Ztru
```

```

e(52,50),Ztrue(52,52),Ztrue(20,10),Ztrue(40,20),Ztrue(30,30),Ztrue(3,40
),Ztrue(30,5),Ztrue(18,50),Ztrue(60,1),Ztrue(8,8),Ztrue(60,60),Ztrue(60
,45),Ztrue(50,30),Ztrue(17,17)];
% % TSampIdx holds the i,j indices of the random test samples
figure(2)
scatter3(Xsensor, Ysensor, Tsensor, 50, Tsensor, 'filled');
view(0,90)

tempIDW = zeros(Nrows,Ncols); %interpolated temperature
for i = 1 : Nrows
    for j = 1 : Ncols
        for n = 1 : Nsamp
            r(n) = sqrt(( Xsensor(n)-Xtrue(i,j) )^2 + ( Ysensor(n)-
Ytrue(i,j) )^2 );
        end
        R = sort(r);
        z = p;
        x = z;
        while R(z+1) == R(z)
            x = z+1; % x= new number of neighbors
            z = z+1;
        end
        rnearest = R(1:x);
        d=zeros(1,x);
        for m = 1 : x;

            if rnearest(m)==0
                d(1,m) = 1;

            else
                d(1,m) = rnearest(m); % test point distance from gauges
                d;
            end
        end
        I = ones(1,x)./d; % inverse distance matrix
        for k = 1 : x
            for q = 1 : Nsamp
                r(q) = sqrt(( Xsensor(q)-Xtrue(i,j) )^2 + ( Ysensor(q)-
Ytrue(i,j) )^2 );
                if r(q)==rnearest(k)
                    Tnearest(k) = Tsensor(q);
                end
            end
        end
        tempIDWm(i,j) = sum (Tnearest(1:x).*I)/sum(I);
    end
end
figure(4)
surf(Xtrue,Ytrue,tempIDWm)
color bar
view(0,90)
title('IDW');

```

Truncated Gaussian Filter:

```
clc
clear
close all

%-----Specification of the grid-----
----
dx = 0.01; %----x spacing
dy = 0.01; %----y spacing
imax = 101; %----max # of points in x direction
jmax = 101; %----max # of points in y direction
%-----# of total cells = (imax-1)*(jmax-1)
%-----Construction-----

for i = 1:imax
    for j = 1 : jmax
        x(i,j) = (i-1)*dx;
        y(i,j) = (j-1)*dy;
    end
end

%-----
%-----
%-----

%-----
%-----Random placement of sensors-----
%-----

N = imax*jmax;
M = 20; %-----total number of sensors
s(1) = unidrnd(N); %-----location of first sensor
i = 2;
while i<=M
    R(i) = unidrnd(N);
    while (min(abs(s(1:i-1)-R(i)))==0)
        R(i) = unidrnd(N);
    end
    s(i) = R(i);
    i = i + 1;
end

%-----
%-----Random temperature assignment-----
%-----

temp=zeros(imax,jmax);

for i = 1:M
    temp(s(i)) = unifrnd(-1,1);
end

%-----
%-----R_p estimation-----
-----
```

```

%-----
alpha = 3;
ww = (1-exp(-alpha))/alpha - exp(-alpha);
I = 20;
NN(1:I-1) = 2*M;
NN(I) = M;
midpoint = (imax*jmax + 1)/2;
testpointlabel = s(1);

for i = 1 : imax
    for j = 1 : jmax
        rpnew = 1;
        for k = 1 : I
            for n = 1 : M
                r(n) = sqrt(( x(s(n))-x(i,j) )^2 + ( y(s(n))-y(i,j) )^2
);
                if r(n)<=rpnew
                    w(n) = exp(-alpha*(r(n)/rpnew)^2)-exp(-alpha);
                else
                    w(n)=0;
                end
            end
            Dp = (sum(w)/ww)/(pi*(rpnew)^2);
            rpnew = sqrt(NN(k)/(pi* Dp));
        end
        T(i,j) = w*(temp(s)')/sum(w);
    end
end

%-----
%-----Interpolated temperature for given coordination-----
Sensotemp = temp(s(1))
rpnew;
%-----
%-----Scatter plot of temperature for all gridded cells-----
figure(1)
scatter3(x(:,50),y(:,50),T(:,50))
title('Mid line');
figure(2)
for j = 1 : jmax
scatter3(x(:,j),y(:,j),T(:,j))
colorbar
hold on
scatter3(x(s),y(s),T(s),100,'filled')
title('Interpolated temperature including Gauge locations')
end
scatter3(x(s),y(s),T(s))
hold off
figure(3)
contour3(T,1000);
colorbar

```

Image Processing (Retrieving Temperature from MODIS images):

```
clc
clear
close all

a=load('june2006latlonstations.txt');
[X, cmap, R1, bbox1] = geotiffread('tjune13v04.tif');
X=double(0.02.*X);
X(X==0) = NaN;
% figure
% geoshow(X, R1, 'DisplayType', 'texturemap');
% colormap(demcmap(X))
% colorbar;
[Y, cmap, R2, bbox2] = geotiffread('tjune13v05.tif');
Y=double(0.02.*Y);
Y(Y==0) = NaN;
figure
geoshow(X, R1, 'DisplayType', 'texturemap');
colormap(demcmap(X))
colorbar;
hold on
geoshow(Y, R2, 'DisplayType', 'texturemap');
colormap(demcmap(Y))
colorbar;
% caxis([300 320])
hold on
scatter(a(:,2),a(:,1))
for i=1 : size(a(:,1))
    if a(i,1) >= bbox1(1,2)
alv4(i,1)=a(i,1);
    else
        alv5(i,1)=a(i,1);
    end
end
indb1=find(alv4);
b1=a(indb1,:);
indb2=find(alv5);
b2=a(indb2,:)

indexj = floor(abs((abs(b1(:,2))-abs(R1(3,1))))/R1(2,1)) + 1;
% index latitude -75:desired lat, R(2,1):dx, bbox(1,1):lat upper left
indexi = floor(abs((abs(b1(:,1))-abs(R1(3,2))))/R1(2,1));
b=zeros(size(b1(:,1)),1);
for i=1: size(indexi(:))
b(i,1) = X(indexi(i), indexj(i));
b(i,1) = ((9/5).*b(i,1))-459.67;
end
belev1=b-0.0204;
belev1withstationindex=zeros(size(belev1),2);
belev1withstationindex(:,1) = belev1;
belev1withstationindex(:,2) = indb1;
belev1withstationindex

index2j = floor(abs((abs(b2(:,2))-abs(R2(3,1))))/R2(2,1)) + 1;
index2i = floor(abs((abs(b2(:,1))-abs(R2(3,2))))/R2(2,1));
```

```
b2a=zeros(size(b2(:,1)),1);
for j=1: size(index2i(:))
b2a(j,1) = Y(index2i(j), index2j(j));
b2a(j,1) =((9/5).*b2a(j,1))-459.67;
end

b2elev = b2a-0.0204;
belev2withstationindex = zeros(size(b2elev),2);
belev2withstationindex(:,1) = b2elev;
belev2withstationindex(:,2) = indb2;
belev2withstationindex
```


Leave one out cross validation:

```
clc
clear
close all
format('shortG')
% error matrix error(station index,Day) , (TG-TM) ,
% error= (Tg-TM)station - (TG-TM)NNstation
[TGplainfield] = xlsread('PLAINFIELD.xls',1,'F2990:F3355');
[TGatlanticcityIntlAp] = xlsread('ATLANTIC CITY INTL
AP.xls',1,'F3022:F3386');
[TGharrison] = xlsread('HARRISON.xls',1,'F3022:F3386');
[TGtomsriver] = xlsread('TOMS RIVER.xls',1,'F2990:F3354');
[TGcharlottesburg] = xlsread('CHARLOTTEBURG RSVR.xls',1,'F2769:F3133');
[TGmillville] = xlsread('MILLVILLE MUNI AP.xls',1,'F3022:F3386');
[TGstellmanor] = xlsread('ESTELL MANOR.xls',1,'F2993:F3357');
[TGnewarlIntlAp] = xlsread('NEWARK INTL AP.xls',1,'F3022:F3386');
[TGcanoebrook] = xlsread('CANOE BROOK.xls',1,'F2865:F3229');
[TGhightstown] = xlsread('HIGHTSTOWN 2 W.xls',1,'F3022:F3386');
[TGtuckerton] = xlsread('TUCKERTON 2 NE.xls',1,'F444:F808');
[TGcranford] = xlsread('CRANFORD.xls',1,'F2991:F3355');
[TGmoorestown] = xlsread('MOORESTOWN.xls',1,'F2959:F3323');
[TGseabrook] = xlsread('SEABROOK FARMS.xls',1,'F3022:F3386');
[TGflemington] = xlsread('FLEMINGTON 5 NNW.xls',1,'F3022:F3386');
[TGfreehold] = xlsread('FREEHOLD-MARLBORO.xls',1,'F2613:F2977');
[TGhammonton] = xlsread('HAMMONTON 1 NE.xls',1,'F728:F1092');
[TGsussex2] = xlsread('SUSSEX 2 NW.xls',1,'F2867:F3231');
[TGnewbrunswick] = xlsread('NEW BRUNSWICK 3 SE.xls',1,'F2991:F3355');
[TGcapemay] = xlsread('CAPE MAY 2 NW.xls',1,'F2958:F3322');
[TGindianmills] = xlsread('INDIAN MILLS 2 W.xls',1,'F3022:F3386');
[TGwertsville] = xlsread('WERTSVILLE 4 NE.xls',1,'F2048:F2412');
[TGessex] = xlsread('ESSEX FELS SVC BLDG.xls',1,'F2014:F2378');
TGplainfield(TGplainfield==99999)=NaN;
TGatlanticcityIntlAp(TGatlanticcityIntlAp==99999)=NaN;
TGharrison(TGharrison==99999)=NaN;
TGtomsriver(TGtomsriver==99999)=NaN;
TGcharlottesburg(TGcharlottesburg==99999)=NaN;
TGmillville(TGmillville==99999)=NaN;
TGstellmanor(TGstellmanor==99999)=NaN;
TGnewarlIntlAp(TGnewarlIntlAp==99999)=NaN;
TGcanoebrook(TGcanoebrook==99999)=NaN;
TGhightstown(TGhightstown==99999)=NaN;
TGtuckerton(TGtuckerton==99999)=NaN;
TGcranford(TGcranford==99999)=NaN;
TGmoorestown(TGmoorestown==99999)=NaN;
TGseabrook(TGseabrook==99999)=NaN;
TGflemington(TGflemington==99999)=NaN;
TGfreehold(TGfreehold==99999)=NaN;
TGhammonton(TGhammonton==99999)=NaN;
TGsussex2(TGsussex2==99999)=NaN;
TGnewbrunswick(TGnewbrunswick==99999)=NaN;
TGcapemay(TGcapemay==99999)=NaN;
TGindianmills(TGindianmills==99999)=NaN;
TGwertsville(TGwertsville==99999)=NaN;
TGessex(TGessex==99999)=NaN;
```

```

TMplainfield=load('plainfield.txt');
TMatlanticcityIntlAp=load('atlanticcityIntlAp.txt');
TMharrison=load('harrison.txt');
TMtomsriver=load('tomsriver.txt');
TMcharlottesburg=load('charlottesburg.txt');
TMmillville=load('millville.txt');
TMestellmanor=load('estellmanor.txt');
TMnewarlIntlAp=load('newarkintlAp.txt');
TMcanoebrook=load('canoebrook.txt');
TMhightstown=load('hightstown.txt');
MTtuckerton=load('tuckerton.txt');
TMcranford=load('cranford.txt');
TMmoorestown=load('moorestown.txt');
TMseabrook=load('seabrook.txt');
TMflemington=load('flemington.txt');
TMfreehold=load('freehold.txt');
TMhammonton=load('hammonton.txt');
TMsussex2=load('sussex.txt');
TMnewbrunswick=load('newbrunswick.txt');
TMcapemay=load('capemay.txt');
TMindianmills=load('indianmills.txt');
TMwertsville=load('wertsville.txt');
TMessex=load('essexfells.txt');

TMplainfielda=load('plainfieldA.txt');
TMatlanticcityIntlApa=load('atlanticcityIntlApa.txt');
TMharrisona=load('harrisonA.txt');
TMtomsrivera=load('tomsriverA.txt');
TMcharlottesburga=load('charlottesburgA.txt');
TMmillvillea=load('millvilleA.txt');
TMestellmanora=load('estellmanorA.txt');
TMnewarlIntlApa=load('newarkintlApa.txt');
TMcanoebrooka=load('canoebrookA.txt');
TMhightstowna=load('hightstownA.txt');
MTtuckertona=load('tuckertonA.txt');
TMcranforda=load('cranfordA.txt');
TMmoorestowna=load('moorestownA.txt');
TMseabrooka=load('seabrookA.txt');
TMflemingtona=load('flemingtonA.txt');
TMfreeholda=load('freeholdA.txt');
TMhammontona=load('hammontonA.txt');
TMsussex2a=load('sussexA.txt');
TMnewbrunswicka=load('newbrunswickA.txt');
TMcapemaya=load('capemayA.txt');
TMindianmillsa=load('indianmillsA.txt');
TMwertsvilla=load('wertsvilleA.txt');
TMessexA=load('essexfellsA.txt');

a= load('xynames.txt');
z=1:183;
location=zeros(size(a(:,1),1),3);
for i=1:size(a(:,1),1)
    j=1;
    location(i,j,z)=a(i,1);
    location(i,2,z)=a(i,2);
end

```

```

locationa=zeros(size(a(:,1),1),3);
for i=1:size(a(:,1),1)
j=1;
locationa(i,j,z)=a(i,1);
locationa(i,2,z)=a(i,2);
end
z=1:183;
location(1,3,z)=TGplainfield(90+z)-TMplainfield(z);
location(2,3,z)=TGatlanticcityIntlAp(90+z)-TMatlanticcityIntlAp(z);
location(3,3,z)=TGharrison(90+z)-TMharrison(z);
location(4,3,z)=TGtomsriver(90+z)-TMtomsriver(z);
location(5,3,z)=TGcharlottesburg(90+z)-TMcharlottesburg(z);
location(6,3,z)=TGmillville(90+z)-TMmillville(z);
location(7,3,z)=TGestellmanor(90+z)-TMestellmanor(z);
location(8,3,z)=TGnewarlIntlAp(90+z)-TMnewarlIntlAp(z);
location(9,3,z)=TGcanoebrook(90+z)-TMcanoebrook(z);
location(10,3,z)=TGhightstown(90+z)-TMhightstown(z);
location(11,3,z)=TGtuckerton(90+z)-MTtuckerton(z);
location(12,3,z)=TGcranford(90+z)-TMcranford(z);
location(13,3,z)=TGmoorestown(90+z)-TMmoorestown(z);
location(14,3,z)=TGseabrook(90+z)-TMseabrook(z);
location(15,3,z)=TGflemington(90+z)-TMflemington(z);
location(16,3,z)=TGfreehold(90+z)-TMfreehold(z);
location(17,3,z)=TGhammonton(90+z)-TMhammonton(z);
location(18,3,z)=TGSussex2(90+z)-TMSussex2(z);
location(19,3,z)=TGnewbrunswick(90+z)-TMnewbrunswick(z);
location(20,3,z)=TGcapemay(90+z)-TMcapemay(z);
location(21,3,z)=TGindianmills(90+z)-TMindianmills(z);
location(22,3,z)=TGwertsville(90+z)-TMwertsville(z);
location(23,3,z)=TGESSEX(90+z)-TMESSEX(z);
z=1:183;
locationa(1,3,z)=TGplainfield(90+z)-TMplainfielda(z);
locationa(2,3,z)=TGatlanticcityIntlAp(90+z)-TMatlanticcityIntlApa(z);
locationa(3,3,z)=TGharrison(90+z)-TMharrisona(z);
locationa(4,3,z)=TGtomsriver(90+z)-TMtomsrivera(z);
locationa(5,3,z)=TGcharlottesburg(90+z)-TMcharlottesburga(z);
locationa(6,3,z)=TGmillville(90+z)-TMmillvillea(z);
locationa(7,3,z)=TGestellmanor(90+z)-TMestellmanora(z);
locationa(8,3,z)=TGnewarlIntlAp(90+z)-TMnewarlIntlApa(z);
locationa(9,3,z)=TGcanoebrook(90+z)-TMcanoebrooka(z);
locationa(10,3,z)=TGhightstown(90+z)-TMhightstowna(z);
locationa(11,3,z)=TGtuckerton(90+z)-MTtuckertona(z);
locationa(12,3,z)=TGcranford(90+z)-TMcranforda(z);
locationa(13,3,z)=TGmoorestown(90+z)-TMmoorestowna(z);
locationa(14,3,z)=TGseabrook(90+z)-TMseabrooka(z);
locationa(15,3,z)=TGflemington(90+z)-TMflemingtona(z);
locationa(16,3,z)=TGfreehold(90+z)-TMfreeholda(z);
locationa(17,3,z)=TGhammonton(90+z)-TMhammontona(z);
locationa(18,3,z)=TGSussex2(90+z)-TMSussex2a(z);
locationa(19,3,z)=TGnewbrunswick(90+z)-TMnewbrunswicka(z);
locationa(20,3,z)=TGcapemay(90+z)-TMcapemaya(z);
locationa(21,3,z)=TGindianmills(90+z)-TMindianmillsa(z);
locationa(22,3,z)=TGwertsville(90+z)-TMwertsvilla(z);
locationa(23,3,z)=TGESSEX(90+z)-TMESSEXA(z);

```

```

t=183;
error=zeros(size(a(:,1),1),t);
for t=1:183;
    for i=1:size(a(:,1),1);
        for j=1 : size(a(:,1),1);

            r(j) = sqrt(( location(i,1,t)-location(j,1,t))^2 + (
location(i,2,t)-location(j,2,t))^2 );
            end
            r=sort(r);

            for m=1:size(a(:,1),1);
                for n=1:size(a(:,1),1);
                    if sqrt(( location(i,1,t)-location(n,1,t))^2 + (
location(i,2,t)-location(n,2,t))^2 )==r(2);
                        m;
                        error(i,t)=location(i,3,t)-location(n,3,t);
                    end
                end
            end

        end

    end
end
for t=1:183;
for i=1:size(a(:,1),1);
    if error(i,t)==min(min(error(:,t)));
        i,t
    end
end
end
errora=zeros(size(a(:,1),1),t);
for t=1:183;
    for i=1:size(a(:,1),1);
        for j=1 : size(a(:,1),1);

            r(j) = sqrt(( locationa(i,1,t)-locationa(j,1,t))^2 + (
locationa(i,2,t)-locationa(j,2,t))^2 );
            end
            r=sort(r);

            for m=1:size(a(:,1),1);
                for n=1:size(a(:,1),1);
                    if sqrt(( locationa(i,1,t)-locationa(n,1,t))^2 + (
locationa(i,2,t)-locationa(n,2,t))^2 )==r(2);
                        m;
                        errora(i,t)=locationa(i,3,t)-locationa(n,3,t);
                    end
                end
            end

        end

    end
end
for t=1:183;
for i=1:size(a(:,1),1);
    if error(i,t)==min(min(error(:,t)));
        i,t
    end
end

```

```

        end
    end
    end
    figure(2)

    for t=1:183;
        scatter(location(:,3,t),location(:,3,t)-error(:,t),'.')
        hold on
        xlabel('TG(i) - TM(i)')
        ylabel('TG(k) - TM(k)')
        title('All Days(terra,tmax)')
        end
        xlim([-70 30])
        ylim([-70 30])
        figure(3)

    for t=1:183;
        scatter(locationa(:,3,t),locationa(:,3,t)-errora(:,t),'.')
        hold on
        xlabel('TG(i) - TM(i)')
        ylabel('TG(k) - TM(k)')
        title('All Days(aqua,tmax)')
        end
        xlim([-70 30])
        ylim([-70 30])
        q=zeros(183,1);
        figure(3)
        for i=1:23;
            for t=1:183;
                q(t,1) = -(error(i,t)-location(i,3,t));
            end
            plot([91:273],q)
            hold on
        end
        for i=1:23;
            for t=1:183;
                if error(i,t)==min(min(error));
                    i,t
                end
            end
        end
    end
    end
    % -----For LOOCV-----
    -----
    % MSE= (sd/sqrt(N))^2
    N=zeros(t,1);
    MSE=zeros(t,1);
    for t=1:t;
        N(t) = size(error(:,t),1)-sum(isnan(error(:,t)),1);
        MSE(t) = ((nanstd(error(:,t)))/sqrt(size(error(:,t),1)-
        sum(isnan(error(:,t)),1)))^2;
    end

    Na=zeros(t,1);
    MSEa=zeros(t,1);
    for t=1:t;
        Na(t) = size(errora(:,t),1)-sum(isnan(errora(:,t)),1);

```

```

MSEa(t) = ((nanstd(errora(:,t)))/sqrt(size(errora(:,t),1)-
sum(isnan(errora(:,t)),1)))^2;
end

```

MODIS enhanced NN interpolation algorithm:

```

clc
clear
close all
% ---Day number for getting cooresponded TG
day=92;
a=load('june2006latlonstations.txt');
[X, cmap, R1, bbox1] = geotiffread('tjune18v04.tif');
X=double(0.02.*X);
X(X==0) = NaN;
[Y, cmap, R2, bbox2] = geotiffread('tjune18v05.tif');
Y=double(0.02.*Y);
Y(Y==0) = NaN;
for i=1 : size(a(:,1))
    if a(i,1) >= bbox1(1,2)
alv4(i,1)=a(i,1);
    else
        alv5(i,1)=a(i,1);
    end
end
indb1=find(alv4);
b1=a(indb1,:);
indb2=find(alv5);
b2=a(indb2,:);
S = shaperead('nj_counties.shp')

p=1;

mstruct = defaultm('tranmerc');
mstruct.geoid = almanac('earth','grs80','survey foot');
mstruct.origin=[38.8333 -74.5 0];
mstruct.falsenorthing = 0.0;
mstruct.falseeasting = 492125.0;
mstruct.scalefactor = 0.9999;
[lat lon] = minvtran(mstruct, [S.X]', [S.Y]');

eEWX=zeros(77,93);
for i=1:76
    for j=1:93
        eEWX(i,j) = (X(i+493,j+1014)-X(i+493,j+1014+1));
    end
end
eEWY=zeros(80,124);
for i=1:80
    for j=1:123
        eEWY(i,j) = (Y(i,j+205)-Y(i,j+205+1));
    end
end
end

```

```

%-----N -S-----
eNSX = zeros(77,93);
for j=1:93
    for i=1:76
        eNSX(i,j) = (X(i+492,j+1014)-X(i+492+1,j+1014));
    end
end
eNSY = zeros(80,124);
for j=1:123
    for i=1:80
        eNSY(i,j)=(Y(i,j+205)-Y(i+1,j+205));
    end
end

EWx=zeros(77*93,1);
for z=1:77*93-1;
    for i=1:77
        for j=1:93
            EWx(z)=eEWX(i,j);
            z=z+1;
        end
    end
end
dfittool(EWx)

EWy=zeros(80*124,1);
for z=1:80*124-1;
    for i=1:80
        for j=1:124
            EWy(z)=eEWY(i,j);
            z=z+1;
        end
    end
end
dfittool(EWy)
EW=zeros(34158,1);
EW(1:14320,1)=EWx;
EW(14321:34158,1)=EWy;

dfittool(EW)

NSx=zeros(77*93,1);
for z=1:77*93-1;
    for i=1:77
        for j=1:93
            NSx(z)=eNSX(i,j);
            z=z+1;
        end
    end
end

NSy=zeros(80*124,1);
for z=1:80*124-1;
    for i=1:80
        for j=1:124
            NSy(z)=eNSY(i,j);

```

```

        z=z+1;
    end
end
end

NS=zeros(34158,1);
NS(1:14320,1)=NSx;
NS(14321:34158,1)=NSy;
nsstd=nanstd(NS)
ewstd=nanstd(EW)
for i=1:size(X,1)
    for j=1: size(X,2)-1
        if(abs(X(i,j)-X(i,j+1))>3.29*ewstd);
            X(i,j)=NaN;
        end
    end
end
for i=1:size(Y,1)
    for j=1:size(Y,2)-1
        if(abs(Y(i,j)-Y(i,j+1))>3.29*nsstd);
            Y(i,j)=NaN;
        end
    end
end
for j=1: size(X,2)
    for i=1:size(X,1)-1
        if(abs(X(i,j)-X(i+1,j))>3.29*ewstd);
            X(i,j)=NaN;
        end
    end
end
for j=1:size(Y,2)-1
    for i=1:size(Y,1)-1
        if(abs(Y(i,j)-Y(i+1,j))>3.29*nsstd);
            Y(i,j)=NaN;
        end
    end
end

[TGplainfield] = xlsread('PLAINFIELD.xls',1,'F2990:F3355');
[TGatlanticcityIntlAp] = xlsread('ATLANTIC CITY INTL
AP.xls',1,'F3022:F3386');
[TGharrison] = xlsread('HARRISON.xls',1,'F3022:F3386');
[TGtomsriver] = xlsread('TOMS RIVER.xls',1,'F2990:F3354');
[TGcharlottesburg] = xlsread('CHARLOTTEBURG RSVR.xls',1,'F2769:F3133');
[TGmillville] = xlsread('MILLVILLE MUNI AP.xls',1,'F3022:F3386');
[TGstellmanor] = xlsread('ESTELL MANOR.xls',1,'F2993:F3357');
[TGnewarlIntlAp] = xlsread('NEWARK INTL AP.xls',1,'F3022:F3386');
[TGcanoebrook] = xlsread('CANOE BROOK.xls',1,'F2865:F3229');
[TGhightstown] = xlsread('HIGHTSTOWN 2 W.xls',1,'F3022:F3386');
[TGtuckerton] = xlsread('TUCKERTON 2 NE.xls',1,'F444:F808');
[TGcranford] = xlsread('CRANFORD.xls',1,'F2991:F3355');
[TGmoorestown] = xlsread('MOORESTOWN.xls',1,'F2959:F3323');
[TGseabrook] = xlsread('SEABROOK FARMS.xls',1,'F3022:F3386');
[TGflemington] = xlsread('FLEMINGTON 5 NNW.xls',1,'F3022:F3386');
[TGfreehold] = xlsread('FREEHOLD-MARLBORO.xls',1,'F2613:F2977');

```



```

[TGhammonton] = xlsread('HAMMONTON 1 NE.xls',1,'F728:F1092');
[TGsussex2] = xlsread('SUSSEX 2 NW.xls',1,'F2867:F3231');
[TGnewbrunswick] = xlsread('NEW BRUNSWICK 3 SE.xls',1,'F2991:F3355');
[TGcapemay] = xlsread('CAPE MAY 2 NW.xls',1,'F2958:F3322');
[TGindianmills] = xlsread('INDIAN MILLS 2 W.xls',1,'F3022:F3386');
[TGwertsville] = xlsread('WERTSVILLE 4 NE.xls',1,'F2048:F2412');
[TGessex] = xlsread('ESSEX FIELDS SVC BLDG.xls',1,'F2014:F2378');

indexj = floor(abs((abs(b1(:,2))-abs(R1(3,1)))/R1(2,1))+1;
% index latitude -75:desired lat, R(2,1):dx, bbox(1,1):lat upper left
indexi = floor(abs((abs(b1(:,1))-abs(R1(3,2)))/R1(2,1)));
b=zeros(size(b1(:,1)),1);
for i=1: size(indexi(:))
b(i,1) = X(indexi(i), indexj(i));
b(i,1) =((9/5).*b(i,1))-459.67;
end
belev1=b-0.0204;
belev1withstationindex=zeros(size(belev1),3);
belev1withstationindex(:,1) = belev1;
belev1withstationindex(:,2) = indb1;

belev1withstationindex(1,3)=TGcanoebrook(day);
belev1withstationindex(2,3)=TGcharlottesburg(day);
belev1withstationindex(3,3)=TGcranford(day);
belev1withstationindex(4,3)=TGessex(day);
belev1withstationindex(5,3)=TGflemington(day);
belev1withstationindex(6,3)=TGfreehold(day);
belev1withstationindex(7,3)=TGharrison(day);
belev1withstationindex(8,3)=TGhightstown(day);
belev1withstationindex(9,3)=NaN;
belev1withstationindex(10,3)=TGnewbrunswick(day);
belev1withstationindex(11,3)=TGnewarlIntlAp(day);
belev1withstationindex(12,3)=TGplainfield(day);
belev1withstationindex(13,3)=TGsussex2(day);
belev1withstationindex(14,3)=TGwertsville(day);
belev1withstationindex
index2j = floor(abs((abs(b2(:,2))-abs(R2(3,1)))/R2(2,1))+1 ;
index2i = floor(abs((abs(b2(:,1))-abs(R2(3,2)))/R2(2,1)));
b2a=zeros(size(b2(:,1)),1);
for j=1: size(index2i(:))
b2a(j,1) = Y(index2i(j), index2j(j));
b2a(j,1) =((9/5).*b2a(j,1))-459.67;
end
b2elev = b2a-0.0204;
belev2withstationindex = zeros(size(b2elev),2);
belev2withstationindex(:,1) = b2elev;
belev2withstationindex(:,2) = indb2;
belev2withstationindex(1,3)=TGatlanticcityIntlAp(day);
belev2withstationindex(2,3)=NaN;
belev2withstationindex(3,3)=NaN;
belev2withstationindex(4,3)=NaN;
belev2withstationindex(5,3)=TGcapemay(day);
belev2withstationindex(6,3)=TGestellmanor(day);
belev2withstationindex(7,3)=TGhammonton(day);
belev2withstationindex(8,3)=TGindianmills(day);

```

```

belev2withstationindex(9,3)=TGmillville(day);
belev2withstationindex(10,3)=TGmoorestown(day);
belev2withstationindex(11,3)=TGseabrook(day);
belev2withstationindex(12,3)=TGtomsriver(day);
belev2withstationindex(13,3)=TGtuckerton(day);

rx=zeros(size(indexi,1),1);
ry=zeros(size(index2i,1),1);

Tintx=zeros(size(X,1),size(X,2));
for i=400:size(X,1);
    for j=800 : size(X,2);

        for z=1:size(indexi,1)
            for z1=1:size(index2i,1)
                rx(z) = sqrt((i-indexi(z))^2+(j-indexj(z))^2);
                ry(z1) = sqrt((569-i+index2i(z1))^2+(j-855-
index2j(z1))^2);
            end
        end
        u=sort(rx);
        ul=sort(ry);

        for zz=1:size(belev1withstationindex,1) ;
            if(sqrt((i-indexi(zz))^2+(j-indexj(zz))^2)==u(1));
                p1=zz;
            end
        end
        for zz3=1:size(belev1withstationindex,1) ;
            if(sqrt((i-indexi(zz3))^2+(j-indexj(zz3))^2)==u(2));
                p7=zz3;
            end
        end
        for zzz=1:size(belev2withstationindex,1) ;
            if(sqrt((569-i+index2i(zzz))^2+(j-855-
index2j(zzz))^2)==ul(1));
                p2=zzz;
            end
        end

        if(u(1)<ul(1))

            if(isnan(belev1withstationindex(p1,3))==1);

                Tintx(i,j)=belev1withstationindex(p7,3)-
belev1withstationindex(p7,1)+(9/5)*X(i,j)-459.67-.0204;
            elseif(isnan(belev1withstationindex(p1,1))==1)
                Tintx(i,j)=belev1withstationindex(p7,3)-
belev1withstationindex(p7,1)+(9/5)*X(i,j)-459.67-.0204;
            else
                Tintx(i,j)=belev1withstationindex(p1,3)-
belev1withstationindex(p1,1)+(9/5)*X(i,j)-459.67-.0204;
            end
        else

```

```

                Tintx(i,j)=belev2withstationindex(p2,3)-
belev2withstationindex(p2,1)+(9/5)*X(i,j)-459.67;
            end
        end
    end

Tinty=zeros(size(Y,1),size(Y,2));
rxx=zeros(size(indexi,1),1);
ryy=zeros(size(index2i,1),1);
for i=1:200;
    for j=1 : 500;
        for z=1:size(index2i,1)
            for z1=1:size(indexi,1)
                ryy(z) = sqrt((i-index2i(z))^2+(j-index2j(z))^2);
                rxx(z1) = sqrt((i+indexi(z1))^2+(j+855-indexj(z1))^2);
            end
        end
        uy=sort(ryy);
        uyl=sort(rxx);

        for zz1=1:size(belev2withstationindex,1) ;
            if(sqrt((i-index2i(zz1))^2+(j-index2j(zz1))^2)==uy(1));
                p3=zz1;
            end
        end
        for zz3=1:size(belev2withstationindex,1) ;
            if(sqrt((i-index2i(zz3))^2+(j-index2j(zz3))^2)==uy(2));
                p8=zz3;
            end
        end

        end
        % -----for uyl>uy
        for zz2=1:size(belev1withstationindex,1) ;
            if(sqrt((i+indexi(zz2))^2+(j+855-indexj(zz2))^2)==uyl(1));
                p4=zz2;
            end

        end

        if(uy(1)<uyl(1))
            if(isnan(belev2withstationindex(p3,3))==1)
                Tinty(i,j)=belev2withstationindex(p8,3)-
belev2withstationindex(p8,1)+(9/5)*Y(i,j)-459.67-.0204;
            elseif(isnan(belev2withstationindex(p3,1))==1)
                Tinty(i,j)=belev2withstationindex(p8,3)-
belev2withstationindex(p8,1)+(9/5)*Y(i,j)-459.67-.0204;
            else
                Tinty(i,j)=belev2withstationindex(p3,3)-
belev2withstationindex(p3,1)+(9/5)*Y(i,j)-459.67-.0204;
            end
        else
            Tinty(i,j)=belev1withstationindex(p4,3)-
belev1withstationindex(p4,1)+(9/5)*Y(i,j)-459.67;
        end
    end
end
end

```

```
end
figure
geoshow(Tintx, R1, 'DisplayType', 'texturemap');
hold on
geoshow(Tinty, R2, 'DisplayType', 'texturemap');
colorbar
hold on
plot(lon,lat)
hold on
scatter(a(:,2),a(:,1),'k*')
title(' 04/02/2006 - MODIS enhanced NN interpolated gauge data
(Geographic coordinate system:NAD83)')
xlabel('Longitude')
ylabel('Latitude')
caxis([60 90])
```

II References

- [1] Widmann, M.L., and Bretherton, C.S., 2000. Validation of mesoscale precipitation in the NCEP reanalysis using a new gridcell dataset for the northwestern United States, *Journal of Climate* 13, pp. 1936-1950.
- [2] Lobell, D.B., Bonfils, C.J., Kueppers, L.M., and Snyder, M.A., 2006. Irrigation cooling effect on temperature and heat index extremes, *Geophysical Research Letters* 35, pp. 210-215.
- [3] Kaplan, J.O and New, M., 2006. Arctic climate change with a 2 degrees C global warming: Timing, climate patterns and vegetation change, *Climatic change* 79. Pp. 213-241.
- [4] Whittaker, R.H., 1967. Gradient analysis of vegetation. *Journal of Biology* 42, pp. 207–264.
- [5] Stephenson, N.L., 1990. Climatic controls on vegetation distribution: the role of the water balance, *American Society of Naturalists* 135, pp. 649–670.
- [6] Cecil, D., Kozłowska, M., 2009. Software sensors are a real alternative to true sensors. *Environmental Modeling Software* 25, pp. 622-625.
- [7] Havlik, D., Schimak, G., and Bleier, T., 2009. Cascading and replicating the OGC Sensor Observation Service, 18th World IMACS / MODSIM Congress, Cairns, Australia 13-17 July 2009
- [8] Douglas, S.K., Klein, R., Goemans, C., and J. Shapiro., 2008. The Impact of Earlier Spring Snowmelt on Water Rights and Administration: A Preliminary Overview of Issues and Circumstances in the Western States. *Western Water Assessment White Paper and Final Project Report*.
- [9] Aberer, K., Hauswirth , M., Salehi, A., (2007), Infrastructure for data processing in large-scale interconnected sensor networks, *Mobile Data Management (MDM)*, Germany, 2007. Available at: <http://lsirpeople.epfl.ch/salehi/papers/GSN-MDM07.pdf>
- [10] Jayasumana, A.P., Han, Q., and Illangasekare, T., 2007. Virtual Sensor Networks - A Resource Efficient Approach for Concurrent Applications, Proc. 4th International Conference on Information Technology: New Generations (ITNG 2007), Las Vegas, NV.

- [11] Ciciriello, P., L. Mottola, and Picco, G.P., 2006. Building virtual sensors and actuators over logical neighborhoods, International Workshop on Middleware for Sensor Networks, MidSens 2006. Co-located with Middleware 2006, Melbourne, 28 November 2006 through 28 November 2006.
- [12] Sanem Kabadayi , S., Christine Julien, C., 2007. Remotely Deployed Virtual Sensors, The Center for Excellence in Distributed Global Environments, The Department of Electrical and Computer Engineering, The University of Texas at Austin. Available at: <http://mpc.ece.utexas.edu/Papers/TR-UTEDGE-2007-010.pdf>
- [13] Serbin, S., and Kucharik, C.J., 2008. Spatiotemporal Mapping of Temperature and Precipitation for the Development of a Multidecadal Climatic Dataset for Wisconsin, *Journal of Applied Meteorological and Climatology* 48, pp. 742-757.
- [14] Willmott, C.J., and Matsuura, K., 1995. Smart Interpolation of Annually Averaged Air Temperature in the United States, *Journal of Applied Meteorology* 34, pp. 2577-2586.
- [15] Willmott, C.J., Rowe, C.M., 1985. Climatology of the Terrestrial Seasonal Water Cycle, *Journal of Climatology* 5, pp. 689-606.
- [16] Isaaks, E.H., and R.M. Srivastava., 1989. *An introduction to applied geostatistics*. Oxford University Press, New York.
- [17] Deutsch, C.V., and Journel, A.G., 1992. *GSLIB:Geostatistical Software Library and User's Guide*. Oxford University Press, New York.
- [18] Hutchinson, M. F., 1991. The application of thin plate splines to continent-wide data assimilation. In: *Data Assimilation Systems*, edited by J. D. Jasper, BMRC Research Report No. 27, pp. 104-113.
- [19] Jarvis, C.H., and Stuart, N., 2000. A Comparison among Strategies for Interpolating Maximum and Minimum Daily Air Temperatures. The Selection of “Guiding” Topographic and Land Cover Variables, *Journal of Applied Meteorology* 40(1), pp. 1060-1074.
- [20] Jarvis, C.H., and Stuart, N., 2000. A Comparison among Strategies for Interpolating Maximum and Minimum Daily Air Temperatures. The Interaction between Number of Guiding variables and the Type of

Interpolation Method, *Journal of Applied Meteorology* 40(2), pp. 1075-1081.

- [21] Thornton, P.E, Running, S.E, and White, M.A., 1997. Generating surfaces of daily meteorological variables over large regions with complex terrain, *Journal of Hydrology* 190, pp. 214-251.
- [22] Hasenauer, H., Merganicova, K., Petritsch, R., Pietsch, S., Thornton, P., 2003. Validating daily climate interpolation over complex terrain in Austria. *Agricultural and Forest Meteorology* 119, pp. 87-107.
- [23] Stahl, K., Moore, r.D., Floyer, J.A., Asplin, M.G., Mckendry, and I.G., 2006. Comparison of approaches for spatial interpolation of daily air temperature in a large region with complex topography and highly variable station density, *Agricultural and Forest Meteorology* 139, pp. 224-236.
- [24] Lundquist, J.D., Pepin, N., and Rochford, C., 2008. Automated algorithm for mapping regions of cold-air pooling in 3 complex terrains, *Journal of Geophysical Research* 113, pp. 1-15.
- [25] Quattrochi, D. A., Luvall, J. C., Rickman, D. L., Estes, M. G., Jr., Laymon, C. A., and Howell, B.F., 2000, A Decision Support Information System for Urban Landscape Management Using Thermal Infrared Data, *Photogrammetric Engineering and Remote Sensing* 66, pp. 1195–1207.
- [26] Nichol, J. and Wong, M.S., 2005. Satellite remote sensing for detailed landslide inventories using change detection and image fusion. *International Journal of Remote Sensing* 26(9), pp. 1913-1926.
- [27] Dousset, B., Gourmelon, F., and Mauri E., 2007. Application of satellite Remote Sensing for Urban Risk Analysis: a case study of the 2003 extreme heat wave in Paris. *Urban Remote Sensing Joint Event*, pp. 11–13.
- [28] Stathopoulou, M., Cartalis, C., 2007. Integrating Corine Land Cover data and Landsat TM for surface emissivity definition: application to the urban area of Athens, Greece. *International Journal of Remote Sensing*. pp. 20-35.

- [29] Deutsch., 2002. *Geostatistical Reservoir Modeling*, Oxford University Press.
- [30] Dodson, R., and Marks, D., 1997. Daily air temperature interpolated at high spatial resolution over a large mountainous region, *Climate Research* 8, pp. 1-20.
- [31] Xia, Y., Febian, P., Winterhalter, M., and Zhao, M., 2001. Forest climatology: estimation and use of daily climatological data for Bavaria, Germany, *Agricultural and Forest Meteorology* 106, pp. 87-103.
- [32] Lookingbill, T.R., and Urban, D.L., 2003, Spatial estimation of air temperature differences for landscape-scale studies in montane environments, *Agricultural and Forest Meteorology* 114, pp. 141-151.
- [33] Jolly, W.M., graham, J.M., Michaelis, A., Nemani, R., and Running, S.W., 2005, A flexible, integrated system for generating meteorological surfaces derived from point sources across multiple geographic scales, *Environmental Modeling & Software* 20, pp. 873-882.
- [34] Garen, D.C., and Marks, D., 2005, Spatially distributed energy balance snowmelt modeling in a mountainous river basin: estimation of meteorological inputs and verification of model results, *Journal of Hydrology* 315, pp. 126-153.
- [35] Degaetono, A.T., and Belcher, B.N., 2007. Spatial Interpolation of Daily Maximum and Minimum Air Temperature Based on Meteorological Model Analyses and Independent Observations, *Journal of Applied Meteorology and Climatology* 46, pp. 1981-1992.
- [36] Hackett, K., and Tsokos, C.P., 2009. A new method for obtaining a more effective estimation of atmospheric temperature in the contiguous United States, *Nonlinear Analysis* 71, pp. 1153-1159.
- [37] Liston, G.E., and Elder, K., A Meteorological Distribution System for High-Resolution Terrestrial Modeling (MicroMet), *Journal of Hydrometeorology* 7, pp. 217-234.

- [38] Wilks, D.S., 2009, A gridded multisite weather generator and synchronization to observed weather data, *Water Resources Research* 45, pp. 1-11.
- [39] Jones, L.A., Ferguson, C.R., Kimball, J.S., Zhang, K., Chan, S.T., McDonald, K.C., Njoku, E.G., and Wood, E.F., 2010, Satellite Microwave Remote Sensing of Daily Land Surface Air Temperature Minima and Maxima From AMSR-E, *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 3(1), pp. 111-123.
- [40] Marquinez J., Lastra J., and Garcia P., 2003. Estimation models for precipitation in mountainous regions: the use of GIS and multivariate analysis. *Journal of Hydrology* 270, pp. 1–11.
- [41] Engel, J.A., Young, G.S., and Hoff, R.N., 2005, Application of Satellite Remote-Sensing Data for Source Analysis of Fine Particulate Matter Transport Events, *Journal of Air & Waste Management Association* 55, pp. 1389-1397.
- [42] Becker, F., and Li, Z.L., 1990. Towards a local split window method over land surface. *International Journal of Remote Sensing* 3, pp. 369-393.
- [43] J. Dozier, J., and Z. Wan, 1994. Development of Practical Multiband Algorithms for Estimating Land Surface Temperature from EOS/MODIS data, *Adv. Space Res* 14, pp.381-389.
- [44] Brutsaert, W., and Cosh, M., 2002, Satellite remote sensing of land surface temperatures: application of the atmospheric correction method and split-window technique to data of ARM-SGP site, *International Journal of Remote Sensing* 23, pp. 5177-5192.
- [45] Sun, Y.Y., Gottsche, F.M., Olesen, F.S., and Fischer, H., 2002, Retrieval of land surface temperature from combined AVHRR data, *Annales Geophysicae* 20, pp. 1257-1260.
- [46] Sun, D., Pinker, T., 2003, Estimation of land surface temperature from a Geostationary Operational Environmental Satellite (GOES-8), *Journal of Geophysical Research* 108(D11), pp. 4326-4341.

- [47] Pinker, R.T., Sun, D., Hung, M., and Li, C., 2009, Evaluation of Satellite Estimates of Land Surface Temperature from GOES over the United States, *Journal of Applied Meteorology and Climatology* 48, pp. 167-180.
- [48] Méndez, A., 2004. Estimate ambient air temperature at regional level using remote sensing techniques. *International Institute for Geo-Information Science and Earth Observation (ITC)*, 86 p.
- [49] Flores, F., Lillo, and M., 2010, Simple air temperature estimation method from MODIS Satellite Images on a regional scale, *Chilean Journal of Agricultural Research* 70(3), pp. 436-445.
- [50] Mahammad, Sk.S., and Ramakrishnan, R., 2003, GeoTIFF - A standard image file format for GIS applications, *Image Processing & Interpretation* 3, pp. 1-25.

