Chapter 4

Modifications to TUF-3D urban micro-climate model to support assessments of WSUD influences on HTC at a micro-scale in urban canyons

4.1 TUF-3D Modifications Design Details

4.1.1 VTUF-3D energy balance modelling with MAESPA tiles

- Modifications to TUF-3D (Krayenhoff & Voogt 2007) to resolve urban canyon radiation flux movement using placeholder vegetation structures which call MAESPA (Duursma & Medlyn 2012) vegetation absorption, transmission, and reflection routines.
- VTUF-3D uses cube shaped structures (as TUF-3D uses to represent buildings) to represent vegetation. These cubes store the surface properties and states and interact with the rest of the VTUF-3D domain.
- The vegetation's true shape is represented in MAESPA and calls underlying MAESPA routines to calculate the vegetation's interactions with the urban canyon and radiation movement (Figure 4.5).
- Using a novel approach, MAESPA tiles replaces VTUF-3D ground surfaces with vegetated MAESPA surfaces and use MAESPA's photosynthesis and water cycle routines to modify VTUF-3D's energy balance calculations.
- Each embedded MAESPA surface calculates a full 3 dimensional tree (along with associated soil and movement of water within the stand) and feeds results back to VTUF-3D ground surface energy balances (Figure 4.9.

4.1.2 Overview of changes

To model the impacts of increased water and vegetation on human thermal comfort in urban areas requires modifications to TUF-3D. TUF-3D in its unmodified form lacks functionality for the physical representation of vegetation along with its physiological processes. Also missing are latent energy fluxes and the water cycle associated with soil, vegetation, evaporation, and precipitation.

Two major changes have been made to the TUF-3D model to add this missing functionality. The first is representation of vegetation's physical form and radiative interactions within an urban canyon. This allows quantifying shading effects and the cooling benefits that can bring. The second is including the physiological workings of vegetation and soil in the model. Being able to model the soil/water, plant and atmosphere interactions allows the role of water to be quantified in urban areas. The entire cycle of soil moisture to vegetation transpiration and evaporation back to precipitation can be examined for its micro-climatic side effects.

In the following section, the three distinct models will be differentiated in the following way. TUF-3D will refer to the unmodified TUF-3D model. MAESPA is the unmodified MAESPA model. VTUF-3D refers to the modified TUF-3D model, including the extracted functionality from MAESPA, which will be the end product of this design document.

4.1.3 TUF-3D unmodified shading logic

TUF-3D sets up a modelling domain by using user specified width/height ratios and calculating a domain of buildings and roads from those. (Other versions exist which allow more control over specific building heights and placements.) Either way leads to a basic starting point of building heights and locations. Two dimensional (x,y) locations of building locations are configured along with their heights (Figure 4.1) and stored in the bldht data object.

These values are used to calculate zH, a roughness height. TUF-3D uses forcing data of temperature, humidity, incoming radiation levels, and wind speed and direction from a location a specified z height (above the canopy). See Appendix (.2.2) for more details. Wind is used in roughness calculations but isn't used to resolve movements around the buildings (as a CFD based model would). This simplification is a design decision trade-off to reduce the complexity of the model and the intensity of computations, while still producing accurate enough results.

The building height values (bldht) are then converted into a 3-dimensional (x,y,z) array (surf_shade). During the model initialization stage, ray tracing is done (Figure 4.2) between all the elements of the surf_shade array to determine which surfaces are visible to from each surface. This step reduces the number of radiative interactions of surfaces which will need to be considered throughout the simulation run.

As the model runs a simulation, at each time step TUF-3D runs a shading routine (Figure 4.3). In it, the model iterates through each surface in the domain (roads and building walls and roofs) and ray-traces towards the sun. Each of these surfaces are divided into quarters and each of these are ray-traced from the centre of each quarter. If the ray passes to the top of the domain without being obstructed (i.e. surface 4) then 0.25 is added to the shade coefficient for that surface. If the ray is obstructed before reaching the top of the domain (i.e. surface 3), then nothing is added. Each surface has potential values of 0, 0.25, 0.5, 0.75 and 1.0. During the later energy balance step, an appropriate amount of incoming radiation is allocated to this surface based on this coefficient.

4.1.4 Adding representations of vegetation

After model improvements are made to VTUF-3D, a similar parallel logic is used to represent the vegetation in the domain. A new user inputted data structure (the veght data structure) is

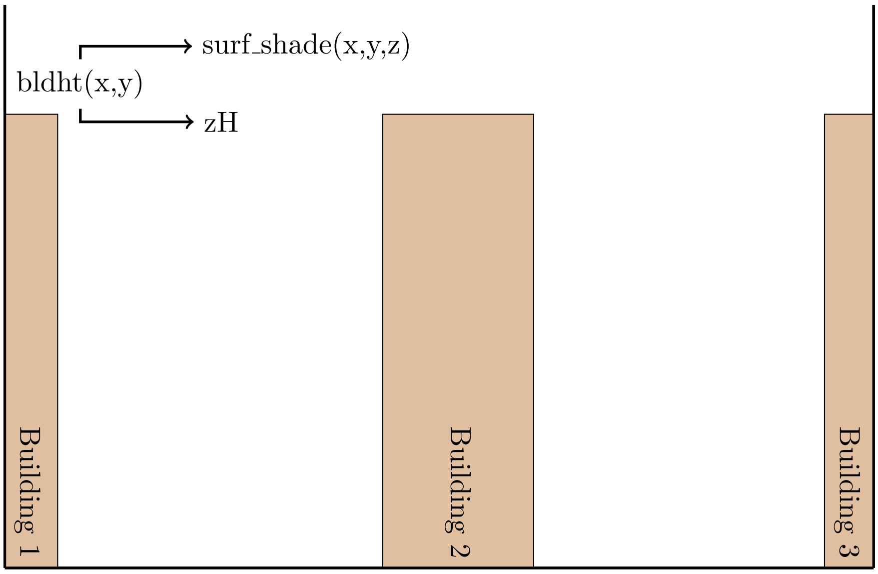

Figure 4.1: *Building height arrays*

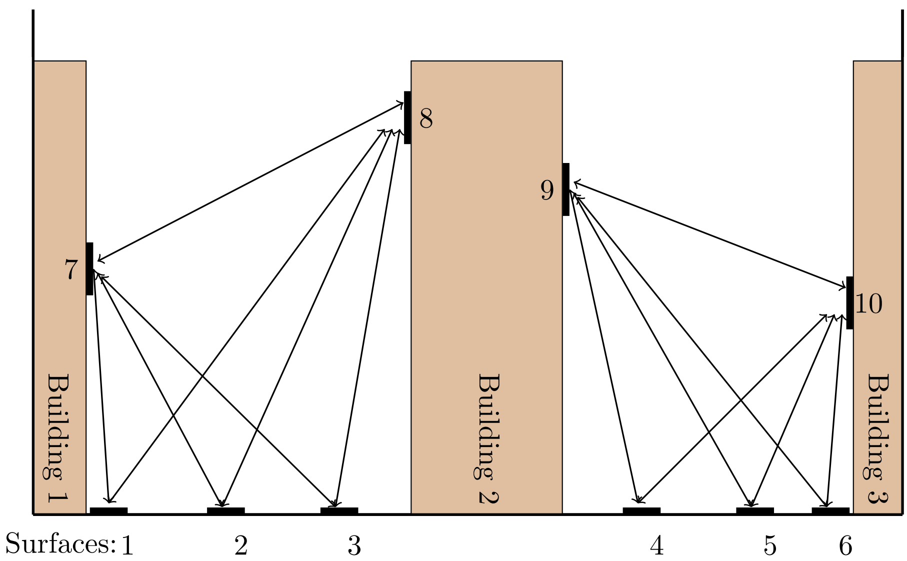

Figure 4.2: *Initial view angles ray tracing*

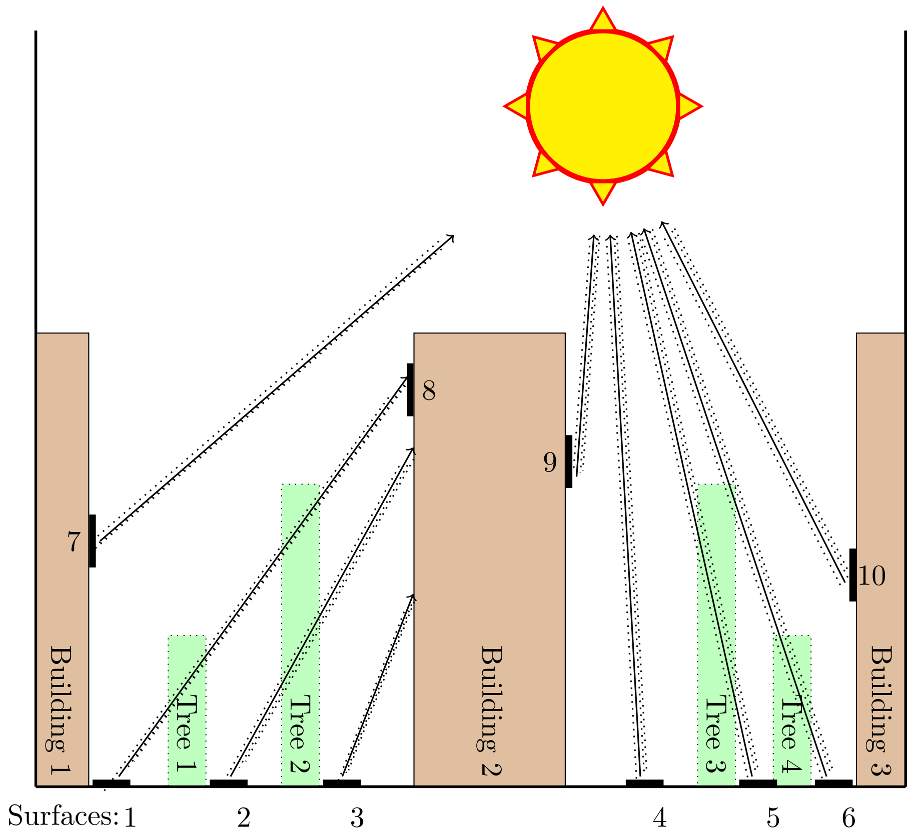

Figure 4.3: *TUF-3D unmodified shading*

added (Figure 4.4) to the model configuration. In this, the (x,y) locations of vegetation in the domain are specified along with their heights. Then like the buildings, this is converted into a 3-dimensional (x,y,z) vegetation shade data structure (vegshade).

The barray_cube() method has been modified to read in the configuration information. Previously, in TUF-3D, this method calculated and laid out a grid of buildings and roads based on a width/height ratio. The new modified version now reads in the domain layout directly from configuration files. These configuration files changes will be described in more detail in Section (4.1.6).

These vegetation elements are ignored in the initializing ray-tracing (Figure 4.2). As this data is used to determine which surfaces will never be visible to each other, vegetation, depending on its density, may allow some level of transmission and must be considered on a case by case basis during the simulation run.

At each time step during the simulation, VTUF-3D now runs a modified shading routine (Figure 4.6). During the model iteration through each surface in the domain, ray-traces towards the sun are done as described above. However, each step of the ray trace checks to see if vegetation has been encountered. If vegetation is encountered, a vegetation encountered flag is set. Then the ray trace continues and concludes when it either passes out of the domain or is blocked by a building.

Table (4.1) summarizes the results for ray tracing the six surfaces of (Figure 4.6). As with the unmodified shading functionality, if a building is found shading the surface, the shading factor is zero, whether vegetation is encountered or not (surfaces 1, 2, 3, and 8) since no

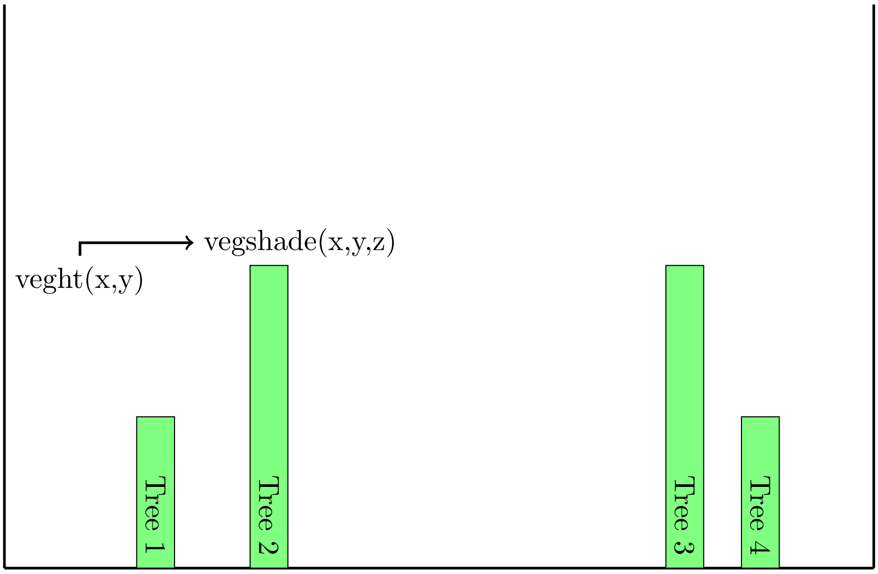

Figure 4.4: *Vegetation height arrays*

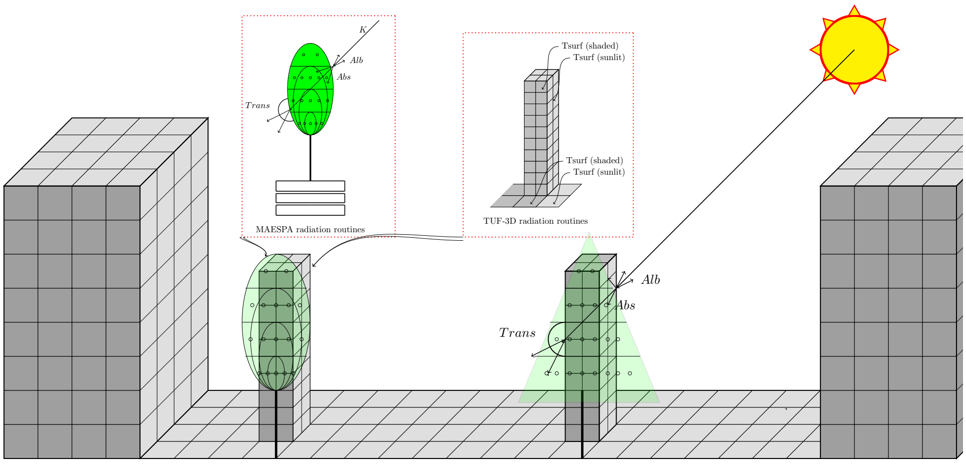

Figure 4.5: *Integration of MAESPA tree model into VTUF-3D radiation fluxes routines*

direct sunlight will reach that surface. If no building or vegetation is encountered, the surface receives the full sunlit factor of 1 (surfaces 4, 7, 9 and 10) (or some faction of that if any of the ray traces from each quarter of the surface encounter an obstructing building). If vegetation is found but the ray leaves the domain otherwise unobstructed, the sunlit factor is currently unknown and requires more processing (surfaces 5 and 6).

To resolve the sunlit factor for surfaces with vegetation, reverse ray tracing is done for those beams (Figure 4.7). For surfaces 5 and 6, ray tracing is done from the top of the domain back along the radiation path. When the ray encounters vegetation (surface 5 and 6), VTUF-3D looks up the tree associated with the ground surface below and calculates the amount of radiation reflected, absorbed, and transmitted through the vegetation (further described in Section 4.1.5).

Table 4.1: *Ray tracing results*

Surface	Sunlit factor	Vegetation in Ray
1	0	True
2	0	True
3	0	False
4	1	False
5	TBD	True
6	TBD	True
7	1	False
8	0	False
9	1	False
10	1	False

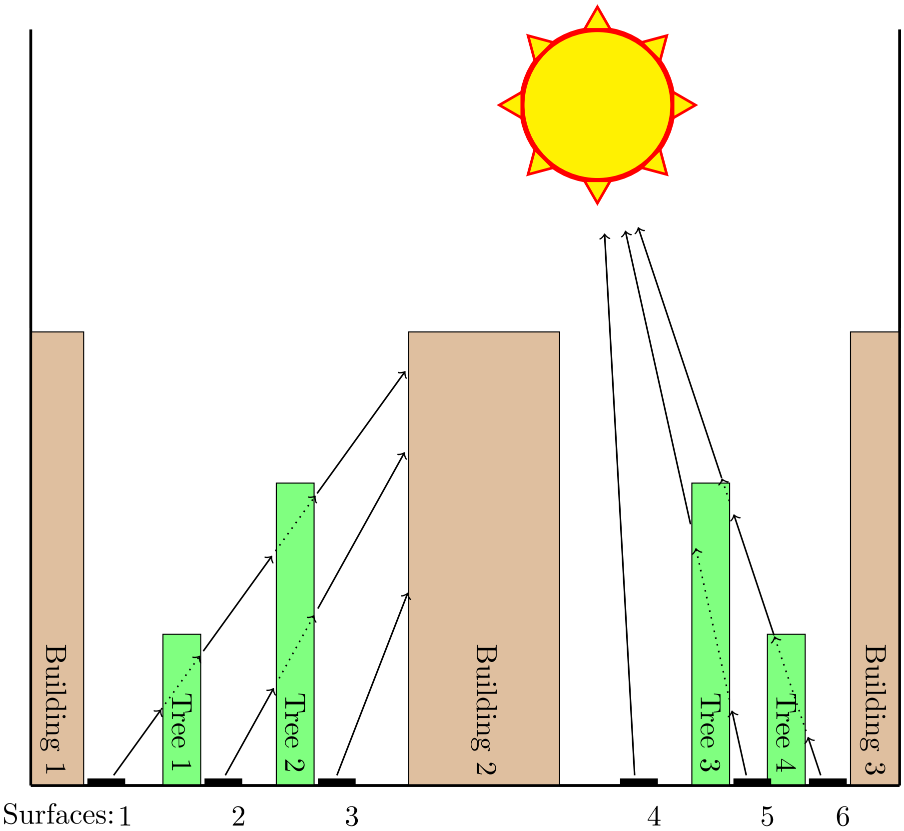

Figure 4.6: *TUF-3D modified shading, timestep ray tracing*

The ray tracing continues, allocating the remaining radiation either to further intercepting vegetation or ultimately to the ground surface. Through this reverse ray tracing process, the problem of inner surface effects is solved by allocating radiation as it enters and transmits through the zero to many objects it encounters along its path.

4.1.5 MAESPA vegetation transmission and energy balance functionality overview

MAESPA, as originally designed allows modelling of a stand of trees or a single grid square containing a single tree. VTUF-3D uses this functionality from MAESPA in two different ways. The first is MAESPA's calculations of radiation movement through the vegetation canopy. Each tree in the VTUF-3D modelling domain is individually modelled by MAESPA before the main modelling run and then this data is extracted by VTUF-3D and used through out the simulation. The configuration of the individual trees will be described in a following section on modelling configuration (Section 4.2.0.5).

VTUF-3D's configuration system has been expanded to allow mapping of individual trees to a (X,Y) location within its domain and to the placeholder vegetation cube and surface structure. These configuration changes (as well as the changes to VTUF-3D configuration) will be further detailed in Section (4.1.6). During the simulation run (described in Section 4.3), calculated transmission values are used to distribute radiation to the appropriate surfaces during the reverse ray tracing process to determine how much radiation is scattered by the vegetation, radiation absorbed by the vegetation, while the remaining amount of radiation is then distributed to subsequent vegetation, and ultimately to the ground surface at the end of the ray.

The second integration of MAESPA is used in the surface energy balances. Flux amounts for each time step are loaded at the beginning of the simulation and used to repartition the energy fluxes for those surfaces with vegetation on them. The logic behind this will be described in Section 4.3.

Through these shading modifications, VTUF-3D stores states of and represents vegetation through placeholder simple block structures in VTUF-3D (Figure 4.8), but calculates the underlying processes using their true shapes and types through the MAESPA representation.

4.1.6 VTUF-3D and MAESPA configuration changes

In its unmodified form, TUF-3D is configured by two main user defined files, *parameters.dat* and *forcing.dat*. These files are described in more detail in Appendix (.2). *Forcing.dat* supplies meteorological forcing data, including incoming shortwave and longwave, air temperature, and vapour pressure, for the duration of the simulation. *Parameters.dat* sets the main properties of the domain being simulated, such as location, orientation, width/height ratios, albedos and emissivities, and other modelling options. These files remain unmodified in their use in VTUF-3D.

In order to add new functionality and modelling options, new files have been added to the MAESPA configuration process. The normal configuration and usage of these files is described in Medlyn & Duursma (2014). To link TUF-3D and MAESPA in VTUF-3D, a new mapping file is introduced, *treemap.dat*. A sample configuration is shown in Appendix (.3.1) and explained in detail in the configuration process (Section 4.2.0.5). All other TUF-3D and MAESPA configuration files remain unchanged from the original models.

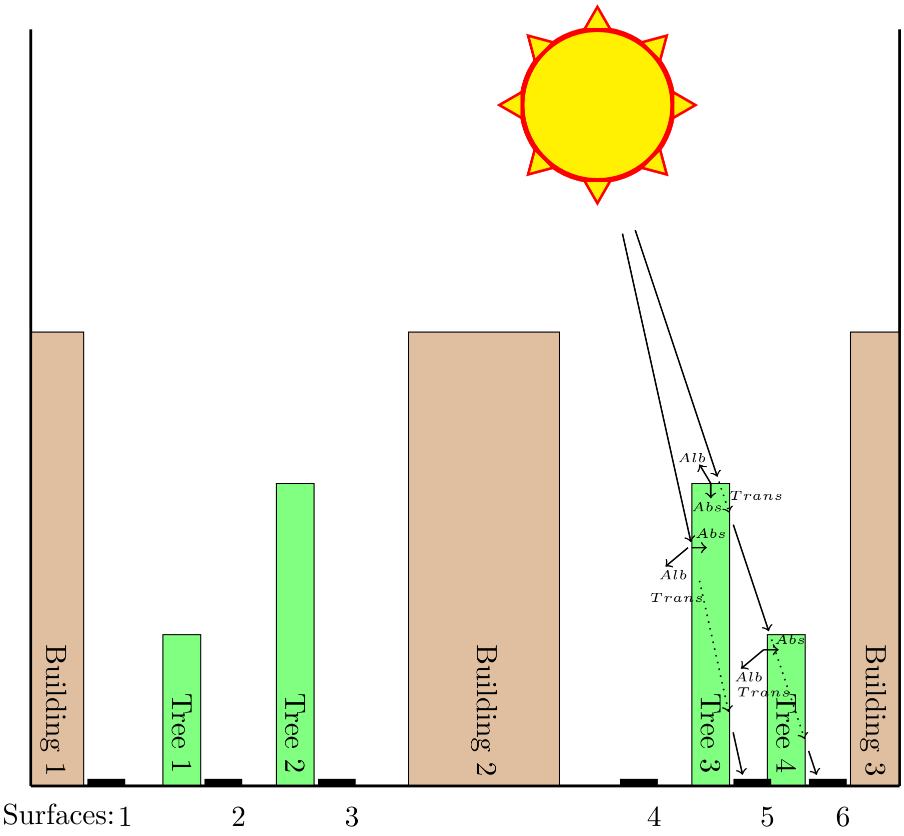

Figure 4.7: *VTUF-3D modified shading, reverse ray tracing*

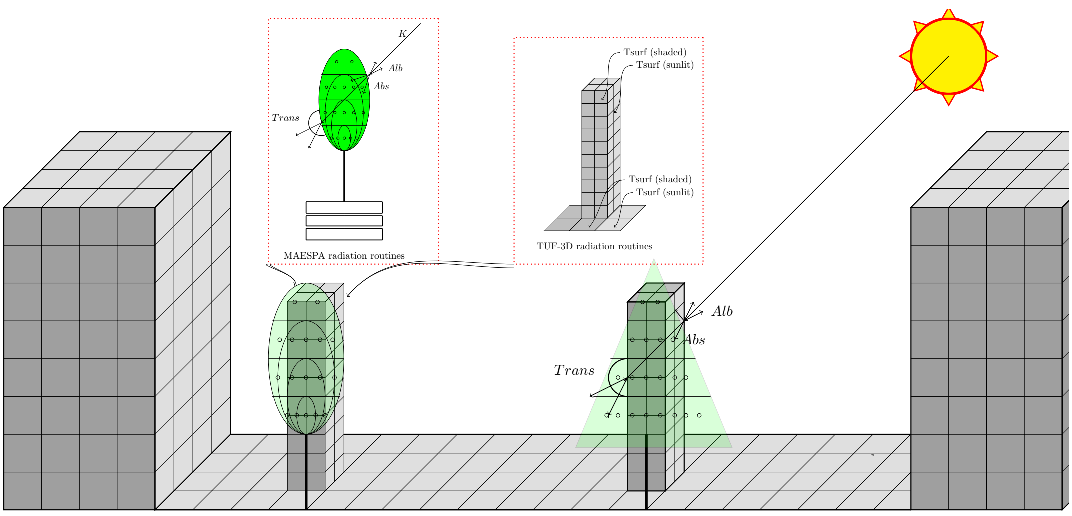

Figure 4.8: *TUF-3D/MAESPA vegetation/radiation interactions*

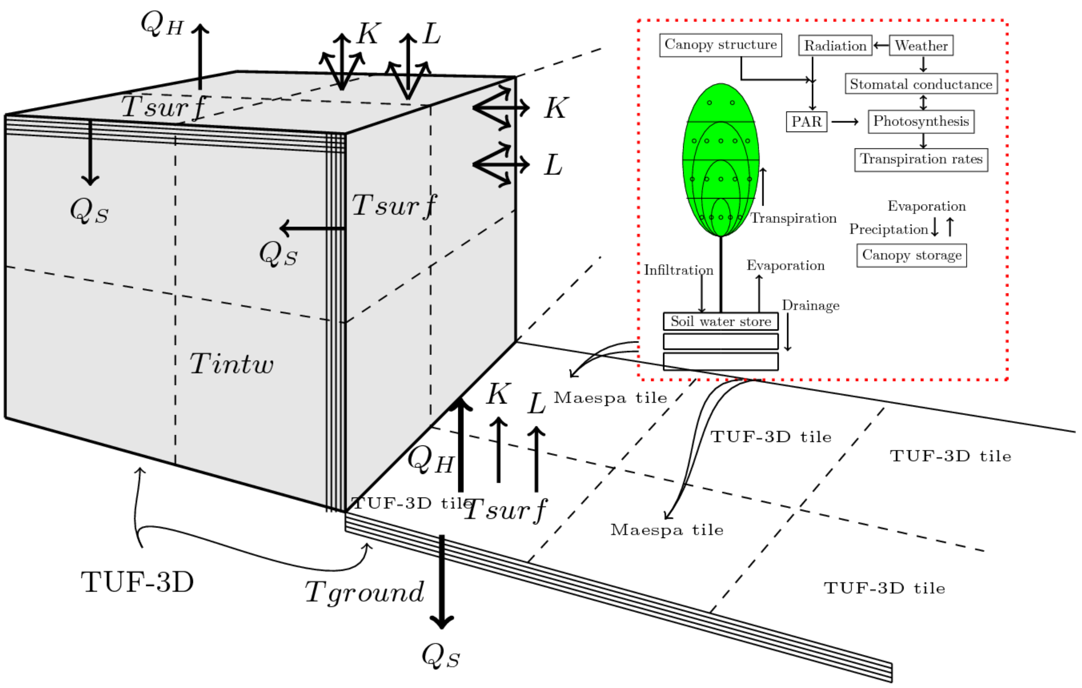

Figure 4.9: *VTUF-3D energy balance modelling with vegetation MAESPA tiles*

4.2 Configuration generation

Overview

In order to simplify the use of the VTUF-3D model, a number of utilities have been created to generate the configuration files needed to run a given simulation. The model will expect to find a number of configuration files and a specific directory structure to organize these files. The general pattern of these directories is:

```
<Run name>
├─1
│  ├─1 <directory containing MAESPA tree 1, diffuse only>
│  ├─2 <directory containing MAESPA tree 1, 50% radiation>
│  └─3 <directory containing MAESPA tree 1, 100% radiation>
├─2
│  ├─1 <directory containing MAESPA tree 2, diffuse only>
│  ├─2 <directory containing MAESPA tree 2, 50% radiation>
│  └─3 <directory containing MAESPA tree 2, 100% radiation>
└─3 <tree directories continue sequentially>
   ├─1
   ├─2
   └─3
```

The overall model configuration, used by CreateMaespaRun, is given in the *messagesConfig.properties* file, setting values for run directory, Julian day start and end, year of run, grid size in meters, and forcing data source.

```
#PBBrush
CreateMaespaRunAndProcess.runDirectory.PBBrush=/home/kerryn/Documents/Work/
  VTUF-Runs/PrestonBase/PrestonBrushbox
```

```
CreateMaespaRunAndProcess.start.PBBrush=40
CreateMaespaRunAndProcess.end.PBBrush=70
CreateMaespaRunAndProcess.gridSize.PBBrush=5
CreateMaespaRunAndProcess.forcing.PBBrush=Preston
CreateMaespaRunAndProcess.year.PBBrush=2004
```

Create streets and buildings

As a first step, four CSV files are read, containing x,y locations and values for building heights, vegetation heights, vegetation types, and a tree map. The heights are multiples of the grid size (i.e. with a 5m grid size, a 10m building's value is 2). Currently, supported vegetation types are:

```
public static final int OLIVE_CONFIG_TYPE = 1;
public static final int NO_TREE_CONFIG_TYPE = 2; //this is grass
public static final int BRUSHBOX_CONFIG_TYPE = 3;
```

Trees are given individual sequential numbers (i.e. 1, 2, 3, etc.), so they can be modelled separately. If a tree extends into another grid square, the grid square of the trunk is given a positive value while the other grid squares containing the canopy are given negative values (i.e. tree 3 might have grid square values of 3, -3, and -3). This allows non-vegetative surfaces to be overshadowed by a canopy.

Create post-processing scripts

A number of scripts are created to be used in the post-processing analysis. In this step, a number of Python and R scripts are created. They will be described in more detail in the post processing step (Section 4.4).

Domain creation

Using the CSV files loaded in Section 4.2.0.2, the *treemap.dat* file is created (and example given in Appendix .3.1).

The number of grid spaces with trees is configured in 'numberTreePlots'. For each of those tree plots, the xLocation, yLocation, phyfileNumber, strfileNumber, treesfileNumber, and treesHeight are specified. The (X,Y) location specifies its place in the domain. MAESPA stores tree properties in the files *phy.dat* (physiology information), *str.dat* (canopy structure), and *trees.dat* (tree characteristics). Using the *treemap.dat* file in VTUF-3D, tree information will be loaded from files indexed by *phyX.dat*, *strX.dat*, and *treesX.dat* for each X value mapped in the *treemap.dat* configuration.

A similar structure in *treemap.dat* configures the buildings in the domain. Number of buildings are given in 'numberBuildingPlots'. For each of those buildings, the xBuildingLocation, yBuildingLocation, and buildingsHeight are specified. The last domain properties values specified in *treemap.dat* are the domain size, 'width' and 'height', in number of grids in each direction.

The overall configuration for VTUF-3D, *parameter.dat*, is created. An example is shown in Appendix .2.1. Forcing data is loaded from a Sqlite database. The query for this database brings back the proper set of data for the time period as well as the given location. An example is shown in Appendix .2.2

MAESPA configurations creation

Each tree is modelled individually using MAESPA, so each of these also need to be configured. The same forcing data is used for each of these as the main modelling run. In order to account for shading of trees by buildings and other trees, each three is modelled three times using modified forcing data. The third tree variation uses unmodified forcing data, the incoming radiation set to 100%. The second tree variation uses 50% of the radiation from the forcing data. The first tree variation uses only diffuse incoming radiation and 0% direct radiation. Each of these variations are put into the directory structure shown in Section 4.2.0.1.

Currently, three tree parametrizations are available, *Olea europaea*, *Lophostemon Confertus*, and grass. Using the templates of parameters described below, configurations of *str.dat* (tree structure parameters), *phy.dat* (physiological parameters), and *tree.dat* (general tree parameters) are created. Some calculations are done to scale some parameters (crown radius, crown height, trunk height, stem diameter, and total leaf area) based on the grid size and user supplied tree height.

Stem diameter is calculated using the relationships described by Buba (2013) and Sumida et al. (2013). If the trunk height plus canopy height is greater than 7m, then stem diameter is calculated in equation (4.2.1) as:

$$stemDiam = ((trunkHeight + crownHeight) - 6.74)/14.4 \tag{4.2.1}$$

Tree physical parameter templates (shown below) are based on a 5m tree. Using the modelled height, the parameters will be scaled according to the logic below. MAESPA uses total leaf area per tree as modelling input, so leaf area index is calculated from the scaled tree dimensions.

```
double oliveZht = 4.0;
double oliveZpd = 1.6;
double oliveZ0ht = 3.0;
double brushboxZht = 10.0;
double brushboxZpd = 3.89;
double brushboxZ0ht = 0.583;

//extent of tree modelling domain
int xmax = gridSizeInMeters;
int ymax = gridSizeInMeters;
//center point of tree modelling domain
double xy = gridSizeInMeters/2;

if (configType == OLIVE_CONFIG_TYPE)
{
        leafAreaIndex = 2.48;
        canopyArea = (crownRadiusX * crownRadiusY) * 3.1415 ;
}
else if (configType == BRUSHBOX_CONFIG_TYPE)
{
        leafAreaIndex = 2.48;
        canopyArea = (crownRadiusX * crownRadiusY) * 3.1415 ;
}
// grass
```

```
else if (configType == NO_TREE_CONFIG_TYPE)
{
        leafAreaIndex = 1.47;
        canopyArea = gridSize*gridSize ;

}
// multiply LAI by the total area of the tree canopy

double totalLeafArea = leafAreaIndex * canopyArea;
```

MAESPA tree parameterization

- Stomatal conductance - Ball-Berry-Opti model (Medlyn et al. 2011)
- nolay = 6 (Number of layers in the crown assumed when calculating radiation interception.)
- pplay = 12 (Number of points per layer)
- nzen = 5 (Number of zenith angles for which diffuse transmittances are calculated.)
- naz= 11 (Number of azimuth angles for which the calculation is done.)

MAESPA olive tree (*Olea europaea*) parameterization

- Tree dimensions for 5x5m grid (rescale for taller/shorter):

  crown radius (m) = 2.5, crown height (m) = 3.75
  trunk height (m) = 1.25, leaf area index=2.48
  crown shape = round, zht (m)=4.0, zpd (m) =1.6, z0ht (m) =3.0
- Leaf reflectance (%PAR, %NIR and %IR) = 0.082, 0.49, 0.05 (Baldini et al. 1997)
- Minimum stomatal conductance g0 (mol/m^{2}s) = 0.0213 (From Smith St. data)
- Slope parameter g1 = 3.018 (From Smith St data)
- # of sides of the leaf with Stomata = 2
- Width of leaf (m) = 0.0102
- CO_2 compensation point (μmol/m ^{2}s) = 46 (Sierra 2012) (56 @ Smith St.)
- Max rate electron transport (μmol/m^{2}s) =135.5 (135.5 @ (Sierra 2012)) (134 @ Smith St.)
- Max rate rubisco activity (μmol/m^{2}s) = 82.7 (82.7 @ (Sierra 2012)) (94 @ Smith St.)
- Curvature of the light response curve =0.9 (Sierra 2012)
- Activation energy of Jmax (KJ/mol) = 35350 (Díaz-Espejo et al. 2006)
- Deactivation energy of Jmax (J/mol) = 200000 (Medlyn et al. 2005)
- XX Entropy term (KJ/mol) = 644.4338
- Quantam yield of electron transport (mol electrons/mol) = 0.2
- Dark respiration (μmol/m^{2}s)= 1.12 (Sierra 2012) (1.79 @ Smith St.)
- Specific leaf area=5.1 (3.65=(Villalobos et al. 1995); 5.1=(Mariscal et al. 2000))

MAESPA brushbox tree (*Lophostemon Confertus*) parameterization

- Tree dimensions for 5x5m grid (rescale for taller/shorter):

  crown radius (m) = 2.5, crown height (m) = 3.75
  trunk height (m) = 1.25, leaf area index = 2.0
  crown shape = round, zht (m)=4.0, zpd (m)=1.6, z0ht (m)=3.0

- Leaf reflectance (%PAR, %NIR and %IR) = 0.04, 0.35, 0.05 (Fung-yan 1999)
- Minimum stomatal conductance g0 (mol/m^{2}s) = 0.01 (Determined from Melbourne Cemetery Tree)
- Slope parameter g1 = 3.33 (Determined from Melbourne Cemetery Tree)
- # of sides of the leaf with Stomata = 1 (Beardsell & Considine 1987)
- Width of leaf (m) = 0.05
- CO_2 compensation point (μmol/m^{2}s) = 53.06 (CO_2 curves)
- Max rate electron transport (μmol/m^{2}s) =105.76 (CO_2 curves)
- Max rate rubisco activity (μmol/m^{2}s) = 81.6 (CO_2 curves)
- Curvature of the light response curve =0.61 (PAR curves)
- Activation energy of Jmax (KJ/mol) = 35350 (Bernacchi et al. 2001)
- Deactivation energy of Jmax (J/mol) = 200000 (Medlyn et al. 2005)
- XX Entropy term (KJ/mol) = 644.4338
- Quantam yield of electron transport (mol electrons/mol) = 0.06 (PAR curves)
- Dark respiration (μmol/m^{2}s) = 1.29 (PAR curves)
- Specific leaf area=25.3 (25.3=(Wright & Westoby 2000))

MAESPA grass parameterization

- Stomatal conductance - Ball-Berry-Opti model (Medlyn et al. 2011)
- Dimensions for grass vegetation for 5x5m grid
  - crown shape = box
  - crown radius (m) = 2.5
  - crown height (m) = 0.1
  - trunk height (m) = 0.1
  - leaf area index=1.47 (Bremer & Ham 2005)
- nolay = 6 (Number of layers in the crown assumed when calculating radiation interception.)
- pplay = 12 (Number of points per layer)
- nzen = 5 (Number of zenith angles for which diffuse transmittances are calculated.)

- naz= 11 (Number of azimuth angles for which the calculation is done.)

The final piece of configuration to generate is the MAESPA *points.dat* files. The use of these is described in Section 4.3.0.11.

The final step in configuration generation is to run each of the MAESPA instances over the period being modelled. This data will then be available for use during the VTUF-3D run.

4.3 Running VTUF-3D

Overview of changes in TUF-3D

There are two major integration points within the newly changed VTUF-3D model. These are in the radiation distribution logic and in the surface energy balance routines. The first item will read vegetation transmission data from MAESPA and use it in the distribution of shortwave radiation and calculation of shading effects. The second item will use MAESPA predictions of latent energy fluxes and water evaporation and vegetation transpiration and use these values to re-partition energy fluxes on those surfaces which contain vegetation. Also, surface temperatures of those surfaces will use values from MAESPA.

Radiation transmission in Maespa

Calculations of radiation transmission are done using the test points functionality from Maespa. This allows modelling of PAR transmissions to user-defined xyz points. The xyz point is half of the grid size (i.e. 2.5 for a 5m grid) and 0.1m to be at the bottom of the tree. This is configured by the *points.dat* configuration file as below:

```
&CONTROL
NOPOINTS = 1
INPUTTYPE = 1
/
&XYZ
COORDS =
 2.5  2.5  0.1
/
&TRANSECT
ANGLE=45
SPACING=0.2
ZHEIGHT=1
/
```

The method getDataForTimeAndDayAndPoint() in the VTUF-3D reverseRayTrace() method reads the hourly output of the test points, the *testflx.dat* data file using the TD, diffuse transmittance to grid point (fraction), value.

VTUF-3D use of Maespa shading values

For each surface, VTUF-3D ray traces four rays (surface is split into quarters) to the sun. As described in Section 4.1.4, if vegetation is found in a ray trace, then a flag is set to also perform a reverse ray trace. The return value from this function is the transmission percentage (0 to 100%) for that ray. Overall then, the sunlit factor for each surface will then be the addition of those four rays, 0 to 4, corresponding giving a range of 0 to 100% illuminated.

```
if (vegetationInRay)then
    !! reverseRayTrace() returns transmissionPercentage for this ray
    call reverseRayTrace()
    sfc(i,sfc_sunlitfact)=sfc(i,sfc_sunlitfact)+transmissionPercentage
    vegetationInRay=.false.
else
    sfc(i,sfc_sunlitfact)=sfc(i,sfc_sunlitfact)+1.
    vegetationInRay=.false.
endif
```

The following code is used to reverse ray trace from the top of the domain (coming from the sun) to the ground surface. As vegetation is encountered, getDataForTimeAndDayAndPoint() is called to read the transmission value for that tree (calculated using the *points.dat* functionality of MAESPA) and reduce the ray strength by that percentage. Tracing continues through any additional vegetation, eventually reaching the ground and allocating the remaining amount of direct shortwave to the ground surface.

```
!! if vegetation is found, reverse ray trace from the top of the domain to find
!! vegetation intersections and distribute sunlit factors to sunlit vegetation
!! and ultimately to the original surface where the forward ray trace originated.
subroutine reverseRayTrace()

!! TUF is 0-24 and Maespa 0-48
maespaHour = int(timeis * 2)
!only calculate transmission through each tree once
lastTreeProcessed = -1
!radiation percentage through each tree
transmissionPercentage = 1.0
!final result after passing through all the trees
finalTransmissionPercentage = 1.0

ztestRev = ZTEST
ytestRev = YTEST
xtestRev = XTEST

xtRev = xt
ytRev = yt
ztRev = zt

xincRev = xinc*(-1.0)
yincRev = yinc*(-1.0)
zincRev = zinc*(-1.0)
if(xincRev.gt.0) xincRev=xincRev*(-1.0)
if(yincRev.gt.0) yincRev=yincRev*(-1.0)
if(zincRev.gt.0) zincRev=zincRev*(-1.0)

!! loop until you reach the ground
DO WHILE (ztestRev.GT.0)
  ztRev=(ztRev+zincRev)
  xtRev=(xtRev+xincRev)
  ytRev=(ytRev+yincRev)
```

```
ztestRev=NINT(ztRev)
xtestRev=NINT(xtRev)
ytestRev=NINT(ytRev)
if (ztestRev.LT.0)then
  exit
ENDIF
if (ytestRev.LT.0)then
  exit
ENDIF
if (xtestRev.LT.0)then
  exit
ENDIF

!! bound check    veg_shade(0:al2+1,0:aw2+1,0:bh+1)
if (xtestRev >al2+1 .or. ytestRev > aw2+1 .or. ztestRev > bh+1) then
  print *,'out␣of␣bounds'
else if (veg_shade(xtestRev,ytestRev,ztestRev))then
  print *,'reverse␣ray␣found␣vegetation␣at␣',xtestRev,ytestRev,ztestRev
  !! find what tree this is
  !! now have tree locations in treeXYMap
  treeConfigLocation=treeXYMap(xtestRev,ytestRev)
  if (treeConfigLocation.eq.-1) then
    print *,'did␣not␣find␣tree␣', xtestRev,ytestRev,ztestRev
  else

  !! at this point, treeConfigLocation gives pointers to the tree
  !! configuration. Calculate transmission, etc and pass it back
  !! to the calling function so it can update sunlit factor
  !! find how much transmits through each tree and get final result
  !!   (only process each tree once)
    if (treeConfigLocation.eq.lastTreeProcessed) then
      !print *,'already processed tree ',treeConfigLocation
    else
      !maespa doesn't have data for non day light hours.
      ! This case shouldn't arise but set it to 0 to make sure
      transmissionPercentage = 0

      !! keep track of how many rays strike the vegetation
      !! at x,y and keep track of the strength of the ray
      !! and how many rays
      treeXYMapSunlightPercentageTotal(xtestRev,ytestRev)
        = treeXYMapSunlightPercentageTotal(xtestRev,ytestRev)
        + finalTransmissionPercentage
      treeXYMapSunlightPercentagePoints(xtestRev,ytestRev)
        = treeXYMapSunlightPercentagePoints(xtestRev,ytestRev)
        + 1

      !! get transmission here
      transmissionPercentage = 1.0
         - getDataForTimeAndDayAndPoint(treeConfigLocation
```

```
                ,0.,maespaHour*1.0,1.,testTDCONST)
        print *,'transmissionPercentage',transmissionPercentage

        lastTreeProcessed = treeConfigLocation
        finalTransmissionPercentage = finalTransmissionPercentage
                                      * transmissionPercentage
      endif
    endif
  endif
end do
```

VTUF-3D use of Maespa differential shading values

Creation of Maespa configurations for each tree creates three different model runs with differing forcing data. The incoming shortwave is varied in three different ways. The first uses 100% values. The second reduces incoming shortwave to 50%. The third uses 0 incoming shortwave.

Two data structures, treeXYMapSunlightPercentageTotal and treeXYMapSunlightPercentage-Points are used to keep track of the amount of direct radiation hitting each tree. This is calculated in reverseRay(), as described in Section 4.3.0.12. When a ray trace finds vegetation, the total for the tree at x,y is increased by the finalTransmissionPercentage. FinalTransmission-Percentage is the cumulative percentage of the incoming shortwave radiation beam. If this beam had previously encountered other vegetation, its current value will be decreased by the new transmission percentage. Also, the number of Points is increased by 1.

```
treeXYMapSunlightPercentageTotal(xtestRev,ytestRev)
  = treeXYMapSunlightPercentageTotal(x,y)
   + finalTransmissionPercentage
treeXYMapSunlightPercentagePoints(xtestRev,ytestRev)
  = treeXYMapSunlightPercentagePoints(x,y) + 1
```

Once these values are used, the ratio of treeXYMapSunlightPercentageTotal to treeXYMapSun-lightPercentagePoints determines which forcing configuration to use. When VTUF-3D uses the values during each timestep, a choice is made which of these sets of data to use with the following logic:

```
diffShadingValueUsed=DIFFERENTIALSHADING100PERCENT
if(Ktot.gt.1.0E-3) then
  if (treeXYMapSunlightPercentagePoints(x,y) .ne. 0) then
     diffShadingCalculatedValue = treeXYMapSunlightPercentageTotal(x,y)
       / treeXYMapSunlightPercentagePoints(x,y)
  else
     diffShadingCalculatedValue=0.
  endif

  if (diffShadingCalculatedValue .gt. .75) then
     diffShadingValueUsed=DIFFERENTIALSHADING100PERCENT
  endif
  if (diffShadingCalculatedValue .le. .75 .and.
          diffShadingCalculatedValue .ge. .25) then
     diffShadingValueUsed=DIFFERENTIALSHADING50PERCENT
  endif
```

```
  if (diffShadingCalculatedValue .lt. .25 ) then
     diffShadingValueUsed=DIFFERENTIALSHADINGDIFFUSE
  endif
endif
```

4.3.1 Adding physiological vegetation processes overview

In order to modify the energy balance calculations of the surfaces and include latent fluxes and water and soil into TUF-3D, VTUF-3D treats the vegetation as ground surface tiles (Figure 4.10). VTUF-3D calculates the energy balances of each surface during each timestep. Incoming radiation energy and existing stored energy is partitioned into new values of storage and radiated heat. In VTUF-3D, a variable for latent energy is added and energy allocated to it from the pool of available energy.

This is done by tiling multiple instances of MAESPA into surfaces which have vegetation. If a surface does not contain vegetation, VTUF-3D runs its normal energy balance calculations. If a surface does contain vegetation, VTUF-3D instead calls a timestep for a self-contained instance of MAESPA. The tiling is done through the *treemap.dat* configuration file, described in Section (4.1.6). This file maps an X,Y grid location to a set of tree configurations and tree heights.

Preceding the energy balance calculations, VTUF-3D calculates, in three dimensions using the canopy structure, incoming radiation (calculated in Section (4.1.5)), and soil and soil water storage.

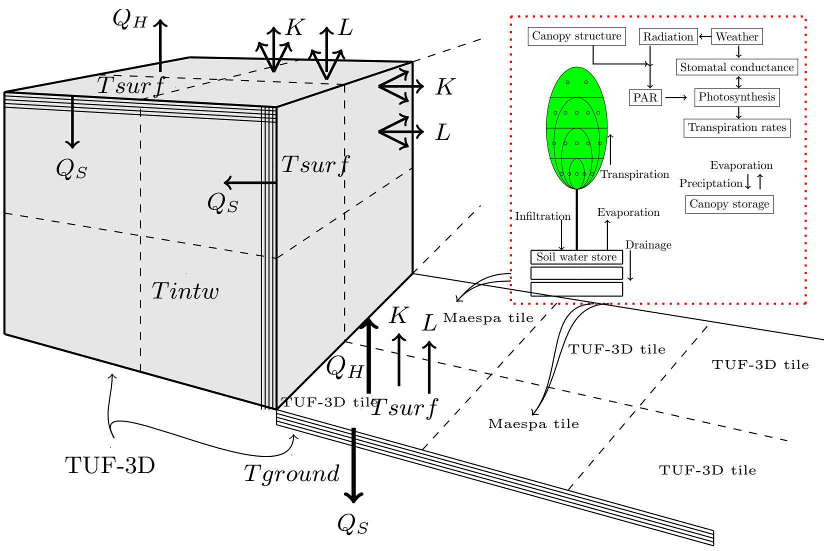

Figure 4.10: *TUF-3D energy balance modelling with MAESPA tiles*

4.3.2 MAESPA energy flux partitioning logic

Loading MAESPA data

As the VTUF-3D model initializes, it loads all of the generated MAESPA data for all the trees and all the different shading configurations.

```
read(WATBALDAT_FILE) watday,wathour,wsoil,wsoilroot,ppt,canopystore,
     evapstore,drainstore,tfall,et,etmeas,discharge,overflow,weightedswp,ktot,
     drythick,soilevap,soilmoist,fsoil,qh,qe,qn,qc,rglobund,rglobabv,radinterc,
```

```
      rnet , totlai , wattair , soilt1 , soilt2 , fracw1 , fracw2 , fracaPAR
read (HRFLXDAT_FILE) DOY, Tree , Spec ,HOUR, hrPAR , hrNIR , hrTHM, hrPs , hrRf ,
      hrRmW, hrLE , LECAN, Gscan , Gbhcan , hrH , TCAN, ALMAX, PSIL , PSILMIN , CI , TAIR , VPD, PAR,
      ZEN, AZ
```

VTUF-3D uses this data in each time step in the simulation. For energy fluxes, VTUF-3D uses the *watbal.dat* and *hrflx.dat* files, the variables in Table 4.2.

Table 4.2: *MAESPA variables loaded*

Variable	Description	Units
canopystore	storage of intercepted rain	mm
evapstore	evaporation of wet canopy	mm
soilevap	soil evaporation	mm
et	modelled canopy transpiration	mm
Tcan	Average foliage temperature	degrees C

After loading, the amount of Qe is calculated using equation 4.3.2.1 and the conversion in Section 4.3.2.2.

$$\begin{aligned} leFromEt = {} & convertMMETToLEWm2(et, treeArea, treeArea, hours)/areaOfTree \\ & + convertMMETToLEWm2(soilevap, treeArea, treeArea, hours)/areaOfTree \\ & + convertMMETToLEWm2(canopystore, treeArea, treeArea, hours)/areaOfTree \\ & + convertMMETToLEWm2(evapstore, treeArea, treeArea, hours)/areaOfTree \end{aligned}$$

MAESPA energy flux conversions

In order to convert mm of evaporation into w/m^2 of latent energy, the following conversion is used.

```
function convertMMETToLEWm2(mm, width1 , width2 , hours)
   real mm
   real convertMMETToLEWm2
   real hours

   !width1,2 are dimensions of plot, x meters, y meters
   !mm is mm of ET
   !18.0152ml/mol of water
   !heat of vaporization 40.7 KJ/mol

   !time (hours) * mm ET * 1M/1000mm * width1 (m) * width2 (m) 10E+06ml/m^3
   !    * 1 mol/18.0152ml * 40.7 KJ/mol *  1W/1000KJ/sec * 60 sec/1 min
   !    * 60 min/hour * hours

   convertMMETToLEWm2=hours * mm * width1 * width2 / 18.0152 * 40.7 * 60*60
end function convertMMETToLEWm2
```

4.3.3 Partitioning in timestep

As the simulation proceeds, during each time step an energy balance is performed on each surface. For each surface with vegetation, the Tsfc is set to the value calculated by MAESPA. For the energy balances themselves, they remain the same calculations as TUF-3D performs on non-vegetated surfaces, but with the following differences.

For *Rnet*, the Rnet value for that surface would have previously be calculated using the albedo and emissivity of the vegetation, set to vegetationAlbedo=0.20 and vegetationEmissivity=0.97 (Oke 1987, p. 12), instead of the TUF-3D surface values. The calculation of upwelling longwave has been deferred until this step, and will complete the calculation of Rnet.

For *Qh*, the Tsfc of the vegetation will yield a different result using the MAESPA Tsfc than if the TUF-3D surface temperature had been used.

For *Qe*, which is a new variable added to VTUF-3D, the value calculated in the MAESPA tree calculations is used.

Finally, *Qg* will be calculated as a residual of $Rnet - Qh - Qe$ instead of the normal TUF-3D method.

```
!use data during each timestep
leFromEt =0

if (treeXYMap(x,y) .ne. 0) then
  tempTimeis = int(timeis*2)
  if (tempTimeis .lt. 1) tempTimeIs = 1
  ! set Tsfc to MAESPA Tcan temp from hrflux
  Tsfc=TCAN+273.15

  !! only use LE from the trunk grid square
  if (treeXYTreeMap(x,y) .gt. 0) then
    leFromEt = qeCalc
  endif
endif

!! if there is vegetation in this grid square
if (treeXYTreeMap(x,y) .gt. 0 ) then
  !! this rnet value would have been calculated previously using the
  !! vegetation alb/emis
  Rnet_tot=Rnet_tot+Rnet-sfc(i,sfc_emiss)*sigma*Tsfc(iab)**4
  Qh_tot=Qh_tot+  ( httc*(Tsfc(iab)-Tconv) )
  Qe_tot=Qe_tot + leFromEt
  !! calculate Qg as a residual from rnet
  Qg_tot=Qg_tot+ (Rnet-sfc(i,sfc_emiss)*sigma*Tsfc(iab)**4)
     - ( httc*(Tsfc(iab)-Tconv) ) - leFromEt
!not vegetation so use TUF-3D's normal method
else
  Rnet_tot=Rnet_tot + Rnet-sfc_emiss*sigma*Tsfc**4
  Qh_tot=Qh_tot + (httc*(Tsfc-Tconv))
  Qe_tot=Qe_tot + leFromEt
  Qg_tot=Qg_tot + (lambda_sfc*(Tsfc-sfc_ab_layer_temp)*2./numlayers )
endif
```

4.4 VTUF-3D Post-processing

Scripts are generated during the configuration generation to complete analysis of the VTUF-3D generated data. These include:

- TUF_ave_graphs.R, used to generates figures for 30 day hourly averages of the $Q*$, Qh, Qe, and Qg fluxes. If applicable, (the run uses Preston forcing data), these fluxes will be compared to observation fluxes.
- TUF_Graphs.R is used to generate numerous figures of every item in the output.
- GenerateUTCIFiles.py is used to read the VTUF-3D output (and forcing data input) and generate UTCI and Tmrt data for each surface in each time step.
- MatlabPatchesPlot3.py is used to create 3D figures of Tsfc predictions. If applicable, (the run uses City of Melbourne Gipps and George St. or Lincoln Square data), points will be annotated on these figures with observation data.
- MatlabPatchesPlot4.py is the same as MatlabPatchesPlot3.py but plots UTCI predictions.
- MatlabPatchesPlot6.py is used to generate 30 day hourly aggregates of predicted VTUF-3D points vs. observations.
- MatlabPatchesPlot8.py is used to create ground level z-slices for each time step of UTCI, Tmrt, and Tsfc predictions.
- UTCI.py is the utility script used by the other scripts to calculate Tmrt and UTCI values.
- TUF_EnergyClosure_graph.R is used to look at the 30 day $Q * -Qh - Qe - Qg = 0$ values to ensure that the model is conserving energy and achieving energy closure.

4.4.1 Tmrt and UTCI

VTUF-3D provides output of downward and upward shortwave and Tsfc for each surface at each time step. Using these and values of air temperature, wind speed, vapor pressure from the forcing data, the values for Tmrt and UTCI are calculated for each surface in the following code.

```
def calcRH(tempC, vapor):
    rh = 100 * vapor / (7.5152E8 * math.exp(-42809/(8.314*(tempC + 273.0))))
    return rh

# calculate mean radiant temperature using Thorsson et al. (2007) scheme

def calcTmrt(kd, ld, ku, lu):
    downAngular = 0.5
    upAngular = 0.5
    AbsorbCoeffSW=0.7
    EmmisHumanBody=0.97
    k = kd * downAngular + ku * upAngular
    l = ld * downAngular + lu * upAngular
    s = AbsorbCoeffSW * k + EmmisHumanBody * l
    tmrt = ( (s / (EmmisHumanBody * 5.67E-8) ) **0.25 ) -273.15
```

```
    return tmrt

#read air temp, wind speed, vapor pressure from forcing data
Ta = forcing[2]
TaK = Ta+273.15
ws = forcing[4]
vapor = forcing[3]
RH=UTCI.calcRH(Ta,vapor)

#read K up and K down from VTUF-3D output for each grid square
kd=kabsDat[j]
ku=kreflDat[j]

#scheme from Offerle et al. (2003) to calculate L up and L down
w=46.5 * vapor / TaK
eClear = 1-(1+w) * exp( -1*(1.2+3.0 * w)**0.5  )
ldn = eClear * 5.67E-08 * TaK**4
groundEmis=0.97
#read Tsfc from VTUF-3D output for each grid square
Tsfc = tsfcDat[j]
TsfcK = Tsfc+273.15
lup = groundEmis * 5.67E-08 * TsfcK**4 + (1-groundEmis) * ldn

Tmrt = UTCI.calcTmrt(kd, ldn, ku, lup)

#UTCI calculation using Broede (2009) formula
utci=UTCI.fUTCI2(Ta, ws, RH, Tmrt)
```

Chapter 5

Validation and assessment of improved performance of the VTUF-3D model to model urban areas

5.1 VTUF-3D Validation Details

5.1.1 Overview of validation process

In order to ensure VTUF-3D can make accurate predictions, an extensive validation process was undertaken. A variety of observation data sets allowed validations of a number of different aspects of the model. A validation matrix (Table 5.1) details the specific validations for each data set. These include validations against observations of air temperature (Ta) and canopy temperatures (Tcan), UTCI human thermal index observations (which include Tmrt observations), evapotranspiration (ET) and tree physiological observations, and flux (energy balance) observations.

Items in table cells colored green have completed validation. Those cells colored yellow are still in progress. Cells colored red have not been started and are still awaiting the observations to be finalized.

Table 5.1: *VTUF-3D validation matrix*

Scenario	Ta	Tcan	UTCI	ET	Energy balance
Preston (Coutts et al. 2007)					
Gipps/George St, Melbourne (Coutts et al. 2015)					
Lincoln Sq, Melbourne (Motazedian 2015)					
Hughesdale					
Smith St, Melbourne (Gebert et al. 2012)					

5.1.2 Model testing and validation using Preston dataset

Validations using the Preston data set were undertaken. Preston is a homogeneous, medium density suburb in the northern part of Melbourne, Australia. The data set contains complete flux observations recorded 2003-2004 (Coutts et al. 2007) from a 30 meter flux tower, allowing validation of surface energy balances against modelled predictions.

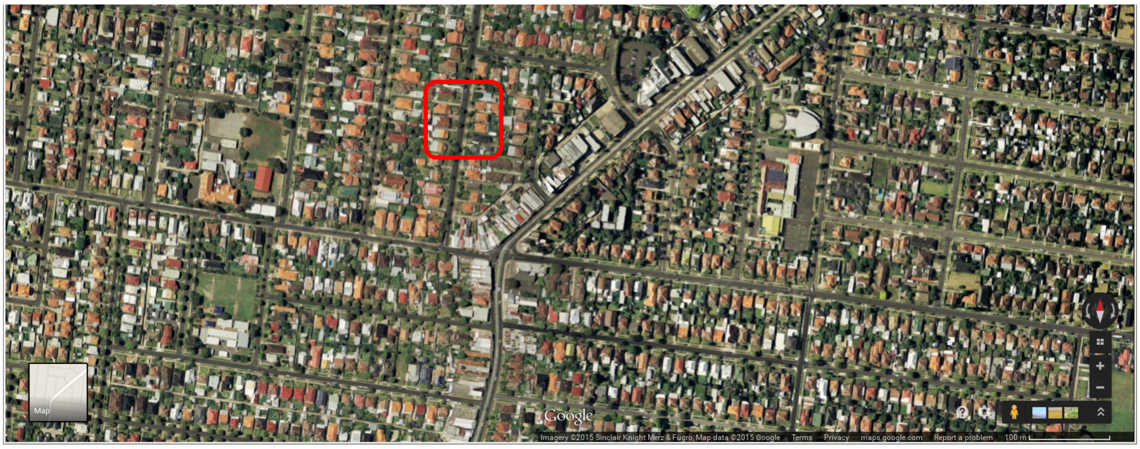

Figure 5.1: *Preston suburb and modelled area (Google 2015)*

The modelled area (500x500m, Figure 5.2) was chosen to be representative of the overall area observed by flux tower. The mix of vegetation types modelled (Figure 5.4), consisting of grass (18.5%), olive (*Olea europaea*) and brushbox (*Lophostemon Confertus*) trees (7.25%), match closely to published values for data set. Modelled building densities (Figure 5.3), consisting of 46.75% buildings and 27.5% impervious surfaces, also match closely to published data set values. Domain resolution is 5m grids.

Figure 5.2: *Digitization of Preston suburban street. (**1**=building heights, **1**=vegetation heights)*

30 days of simulation were run between the dates 10 February 2004 and 10 March 2004, forced by the observations for those days. Individual fluxes ($Rnet$, Qh, Qg, Kdn, and Qe) were

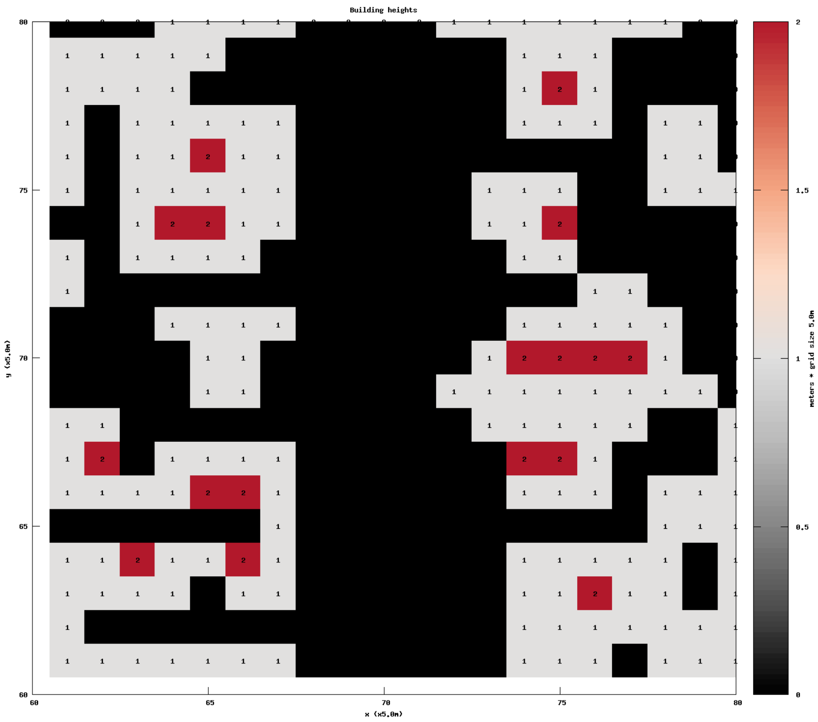

Figure 5.3: *Building heights (0, 5, 10m)*

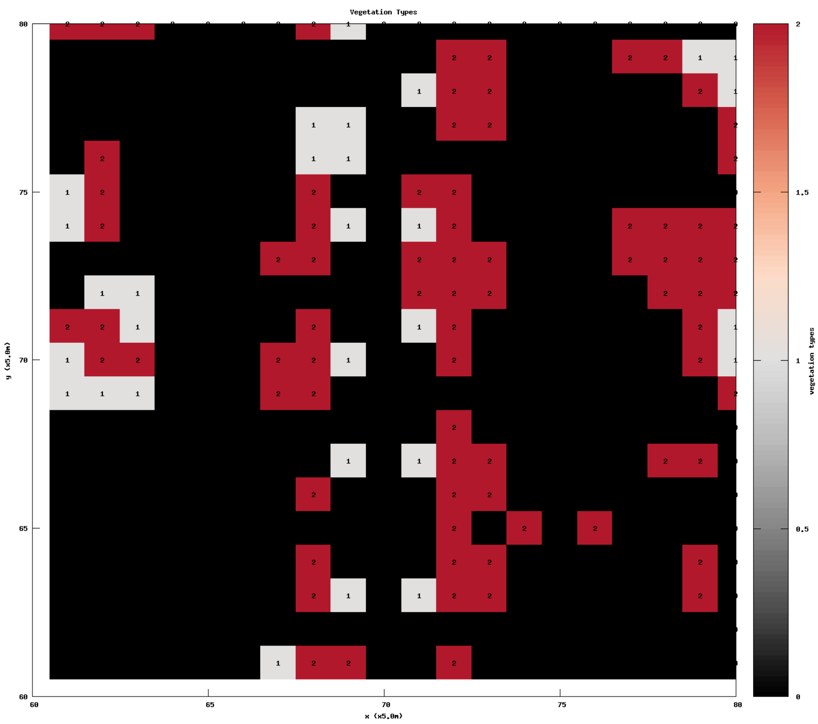

Figure 5.4: *Vegetation heights (0, 5, 10m)*

aggregated into hourly averages over the 30 days and compared to the observations (Figure 5.7).

A number of partitioning schemes were considered during the model development. The following schemes are described in Table 5.2 and overall error performance shown in Figure 5.5 and Figure 5.6.

Table 5.2: *Preston simulations compared*

Name	Description
PrestonBaseNoVeg1	Baseline no-vegetation simulation showing unimproved VTUF-3D
IntercomparisonMax	Grimmond et al. (2010) intercomparison performance maximum (using Grimmond & Oke (1999) VL92 dataset)
IntercomparisonMin	Grimmond et al. (2010) intercomparison performance minimum (using Grimmond & Oke (1999) VL92 dataset)
IntercomparisonMean	Grimmond et al. (2010) intercomparison performance mean (using Grimmond & Oke (1999) VL92 dataset)
PrestonBase6	Section 4.3.3 scheme and Qe equation 4.3.2.1 (divide by tree area) using Olive trees
PrestonBase5	Section 4.3.3 scheme and Qe equation 4.3.2.1 (divide by grid area) using Olive trees
PrestonBrushbox	Section 4.3.3 scheme and Qe equation 4.3.2.1 (divide by tree area) using Brushbox tree
PrestonBrushboxDiff	Section 4.3.3 scheme and Qe equation 4.3.2.1 (divide by tree area) using Brushbox tree differential shading
PrestonBrushboxDiff2	Section 4.3.3 scheme and Qe equation 4.3.2.1 (divide by tree area) using Brushbox tree with differential shading switch off
PrestonValidation13	Section 4.3.3 scheme (but using Maespa Rnet) and Qe equation 4.3.2.1 (divide by grid area) using Olive trees

Runs PrestonBase6 and PrestonBrushbox both use the same scheme described in Section 4.3.3 scheme and Qe equation 4.3.2.1. PrestonBase6 uses olive trees while PrestonBrushbox uses Brushbox trees as the modelled tree species. Both simulations show good energy closure (PrestonBase6 shown in Figure 5.8).

Both simulations show reasonably good agreement between modelled and observed fluxes. 30 day hourly averages for PrestonBase6 (Figure 5.7) show close agreement with a slight over prediction of Qh during the afternoons, an slight under prediction of Qg during late afternoons, and a under prediction of Qe during the day.

Error analysis for PrestonBase6 (Figure 5.9) shows d values of 0.943, 0.556, 0.999, and 0.939 for Qh, Qe, $Q*$, and Qg respectively, and RMSE (w/m^2) values of 56.7, 36,5, 18.5,and 60.6. Daytime and nighttime values (Figures 5.10 and 5.11) do not show a detectable improvement or worsening for those times.

These values show an improvement over the unimproved VTUF-3D model (PrestonBaseNoVeg1, shown in Figures 5.13 and 5.12). The unimproved model yields results of d of 0.902, 0, 0.989, and 0.871 and RMSE (w/m^2) of 90.6, 48.4, 51.9, 77.7 for Qh, Qe, $Q*$, and Qg respectively.

Results of d and RMSE are also compared to results from the Grimmond et al. (2010) intercomparison project in Figures 5.5 and 5.6 and perform within the range of other models. However,

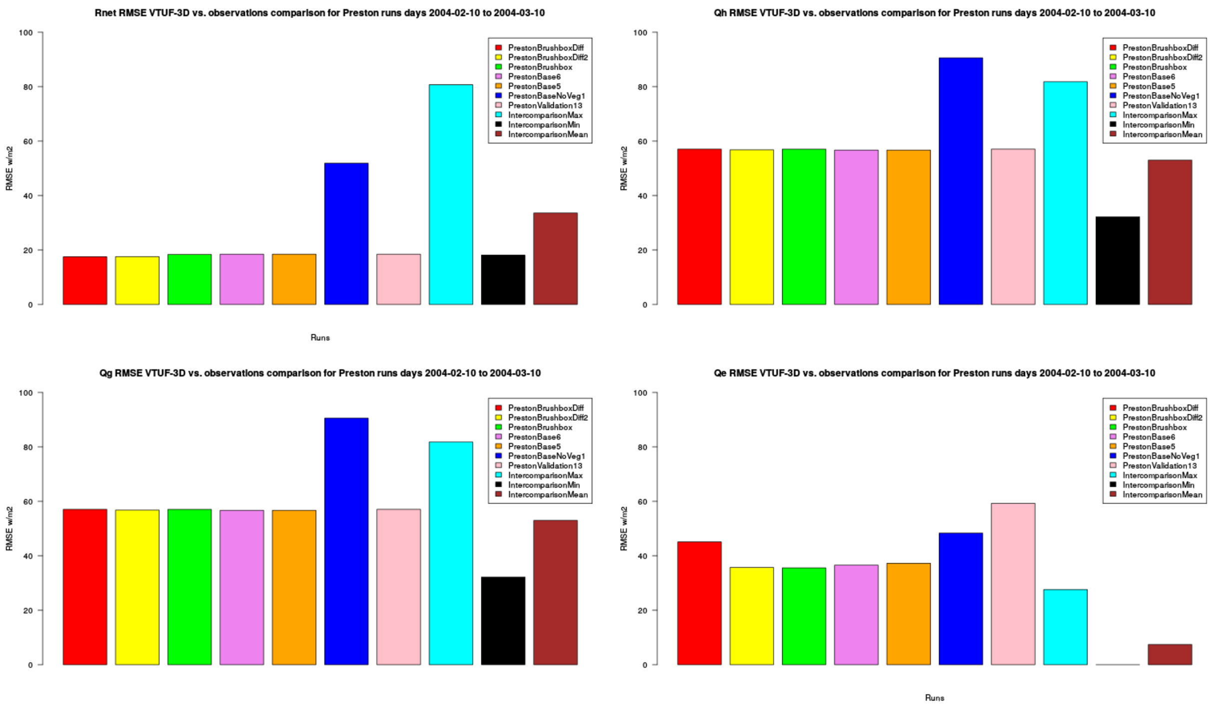

Figure 5.5: *Rnet RMSE VTUF-3D vs. observations comparison for Preston runs*

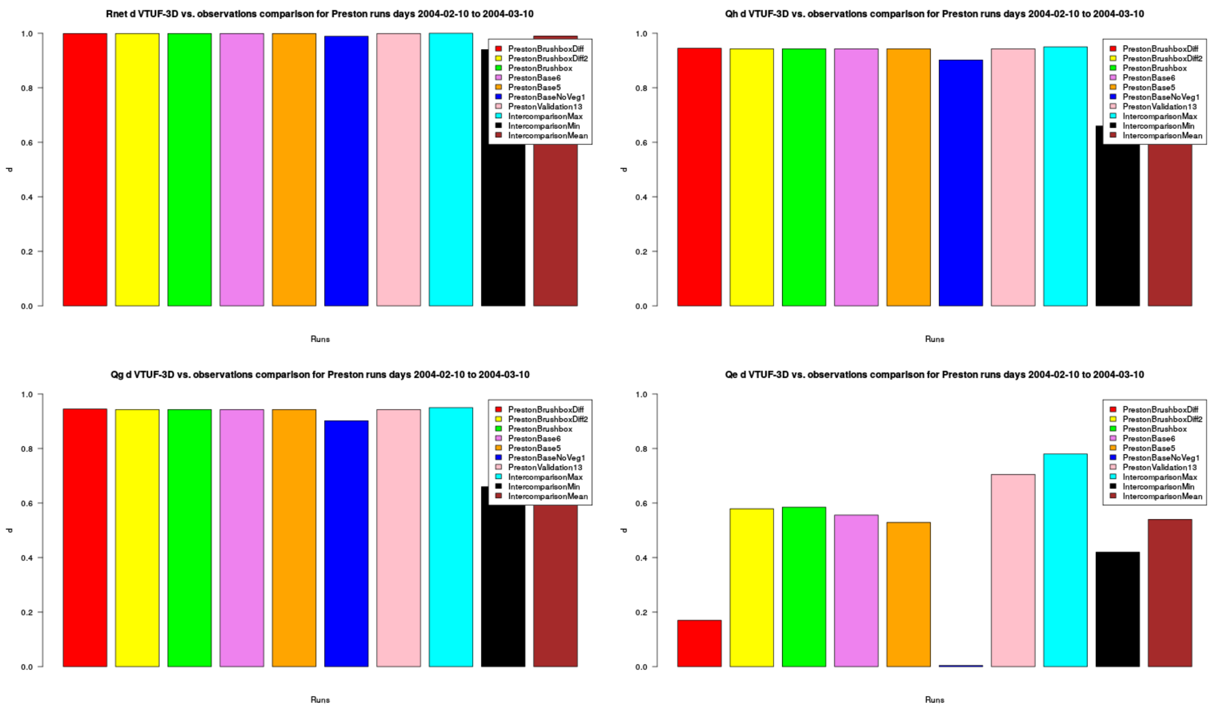

Figure 5.6: *Rnet d VTUF-3D vs. observations comparison for Preston runs*

these results cannot be compared directly as most models in this comparison project were local scaled (not micro-scaled) and these results used a different data set (the Grimmond & Oke (1999) VL92 dataset).

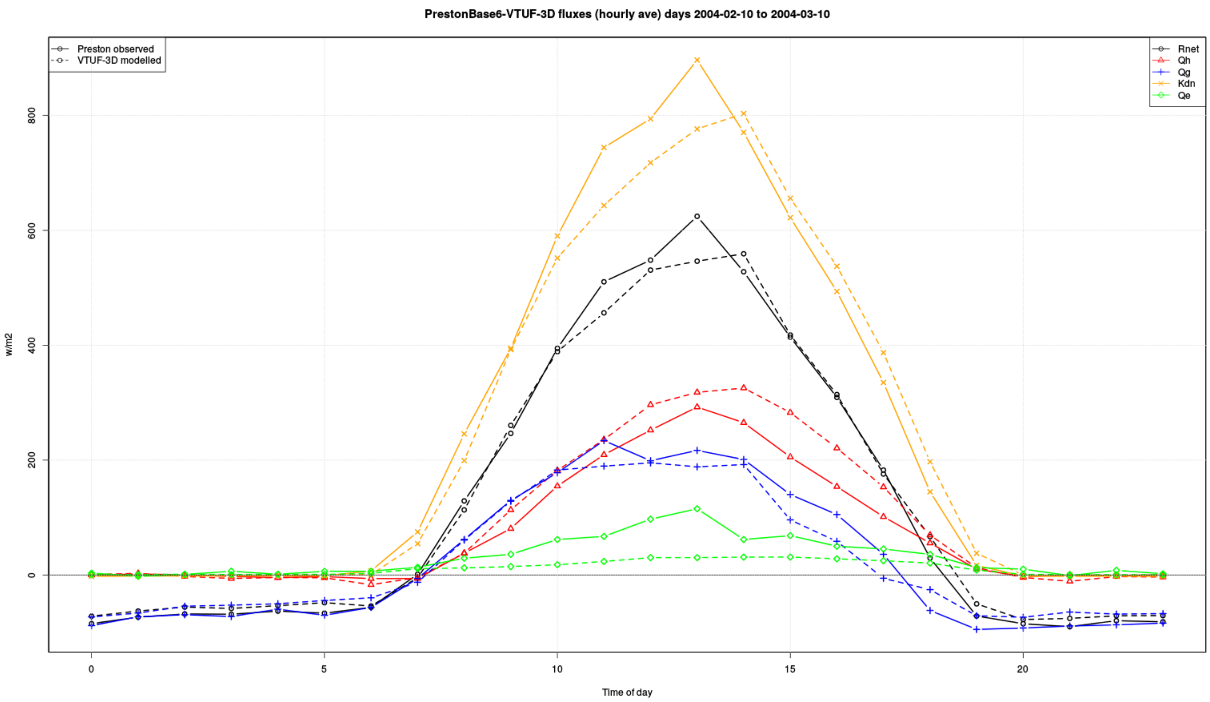

Figure 5.7: *PrestonBase6 run 30 day hourly average VTUF-3D flux comparisons to Preston flux observations*

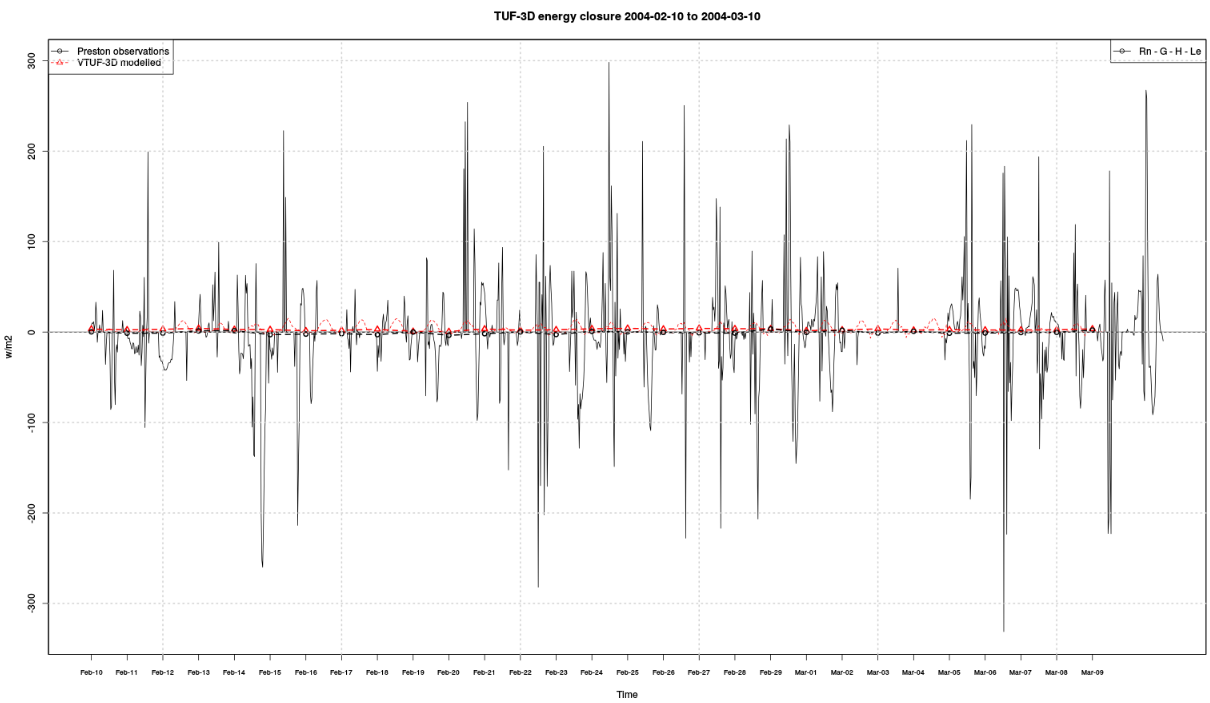

Figure 5.8: *PrestonBase6 energy closure (VTUF-3D and observations),* $Q* - Qg - Qh - Qe = 0$

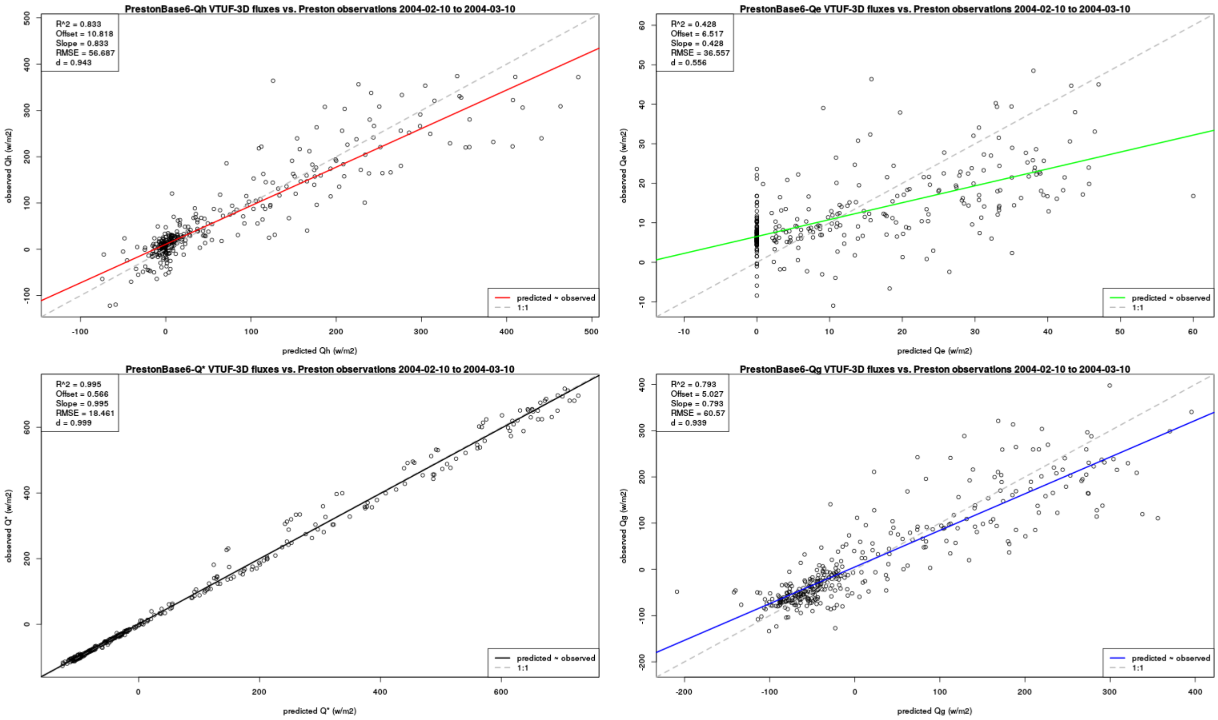

Figure 5.9: *PrestonBase6 predicted vs. observations*

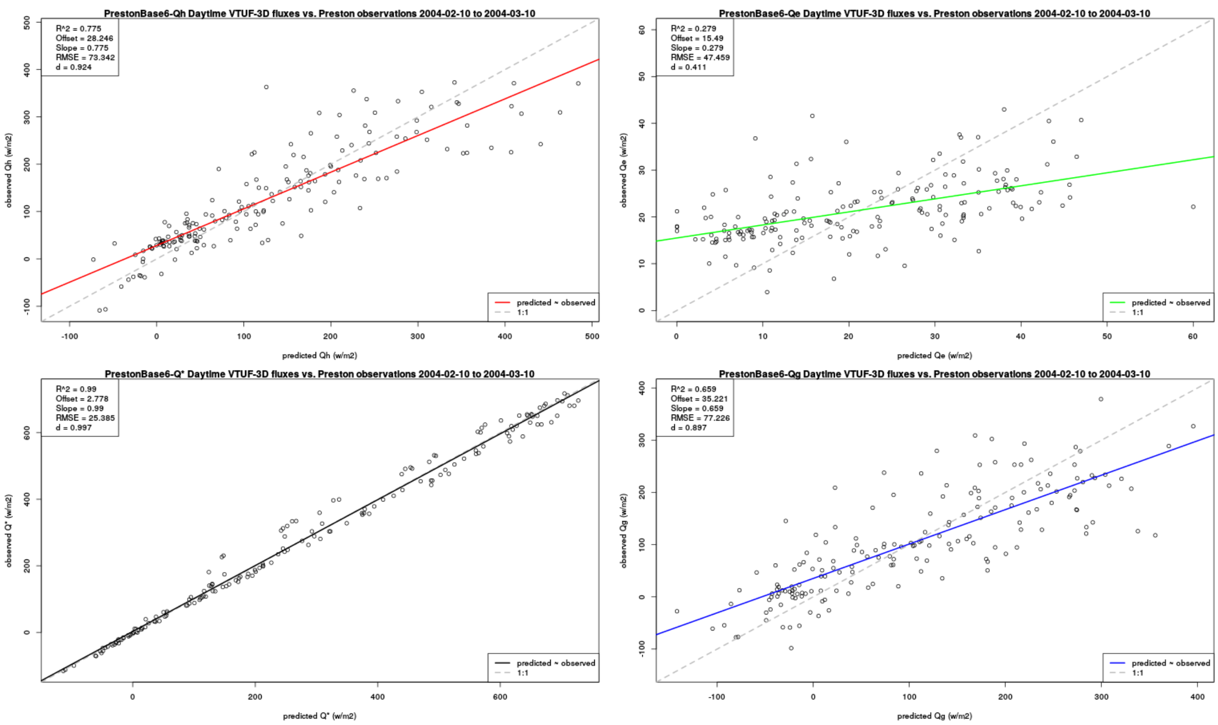

Figure 5.10: *PrestonBase6 daytime predicted vs. observations*

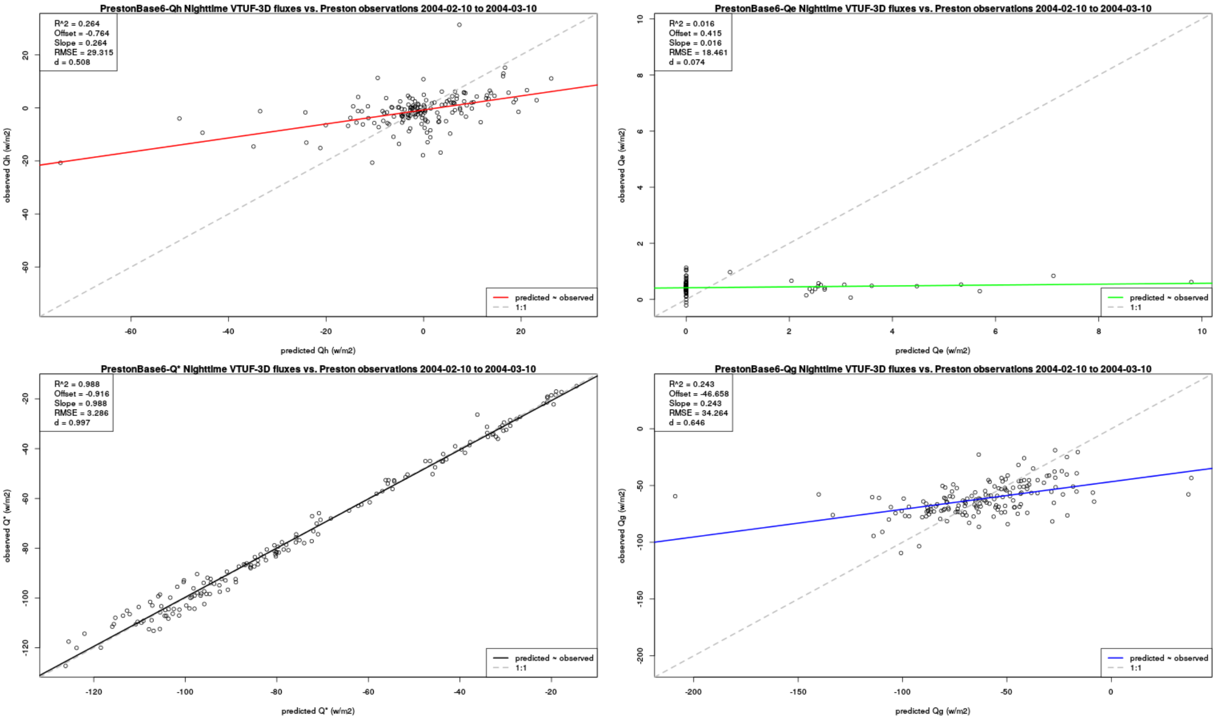

Figure 5.11: *PrestonBase6 nighttime predicted vs. observations*

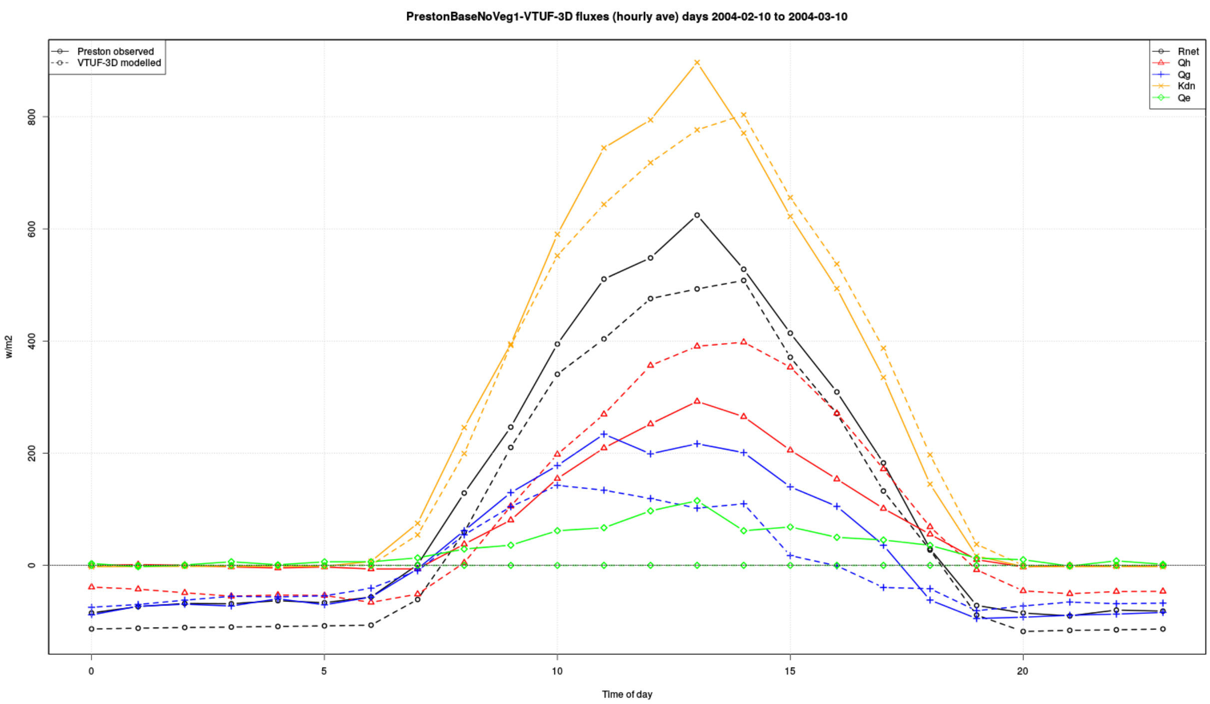

Figure 5.12: *PrestonBaseNoVeg1 run 30 day hourly average VTUF-3D flux comparisons to Preston flux observations*

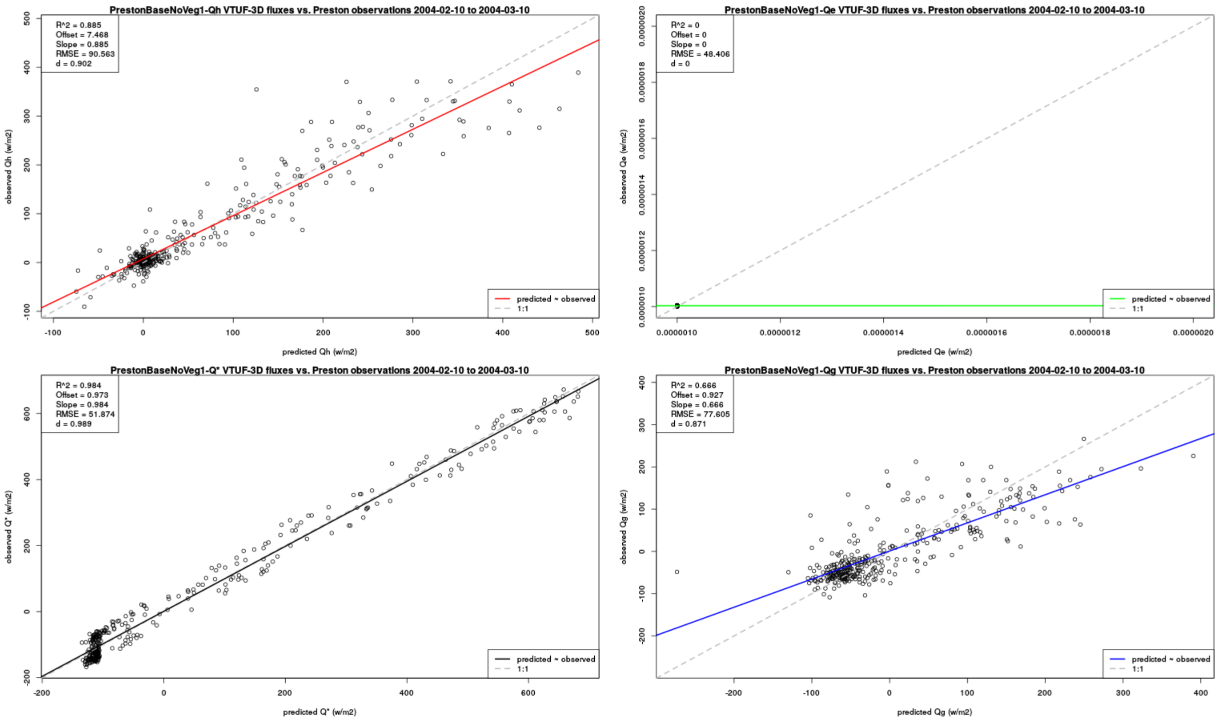

Figure 5.13: *PrestonBaseNoVeg1 predicted vs. observations*

In considering the tree differential shading scheme, these have shown a slight worsening in flux predictions. This scheme yields results of d of 0.945, 0.17, 0.999, and 0.945 and RMSE (w/m^2) values 57.1, 45.1, 17.5, 57,8 for Qh, Qe, $Q*$, and Qg respectively. Also, overall performance of these schemes is shown in Figure 5.5 and Figure 5.6.

5.1.3 Model testing and validation using City of Melbourne, George and Gipps St. dataset

A second set of validations was undertaken using observations from George St. and Gipps St. in the City of Melbourne (Coutts et al. 2015). These observations were taken at a number of observation stations, recording air temperature, wind speed, humidity, and incoming short wave, located along the two streets. Observation data set also contains values for Tmrt and UTCI for each observation station.

Station locations, noted on Figure 5.14 as yellow pins (and station name, EM12, EM9, etc.), also shows the modelled domains. Both streets (George St. and Gipps St.) are shallow urban canyons (average building heights 7 and 8m, H:W 0.32 and 0.27) with varying canopy cover (45% and 12%). These domains were configured accordingly (Figure 5.15). The model was run for both streets for the period 1 February 2014 to 1 March 2014. Domain resolution is 5m grids.

5.1.4 Model testing and validation using George St. dataset

Energy closure for George St. (Figure 5.16) shows some energy gain during the modelled period. Further work will address this issue.

In a point by point comparison, using 30 day hourly averages (Figure 5.21), the model over predicts Tmrt during the day and under predicts it at night. This leads to UTCI, which is

Figure 5.14: *George St/Gipps St - Modelled domains with 4 and 3 observation stations located on street*

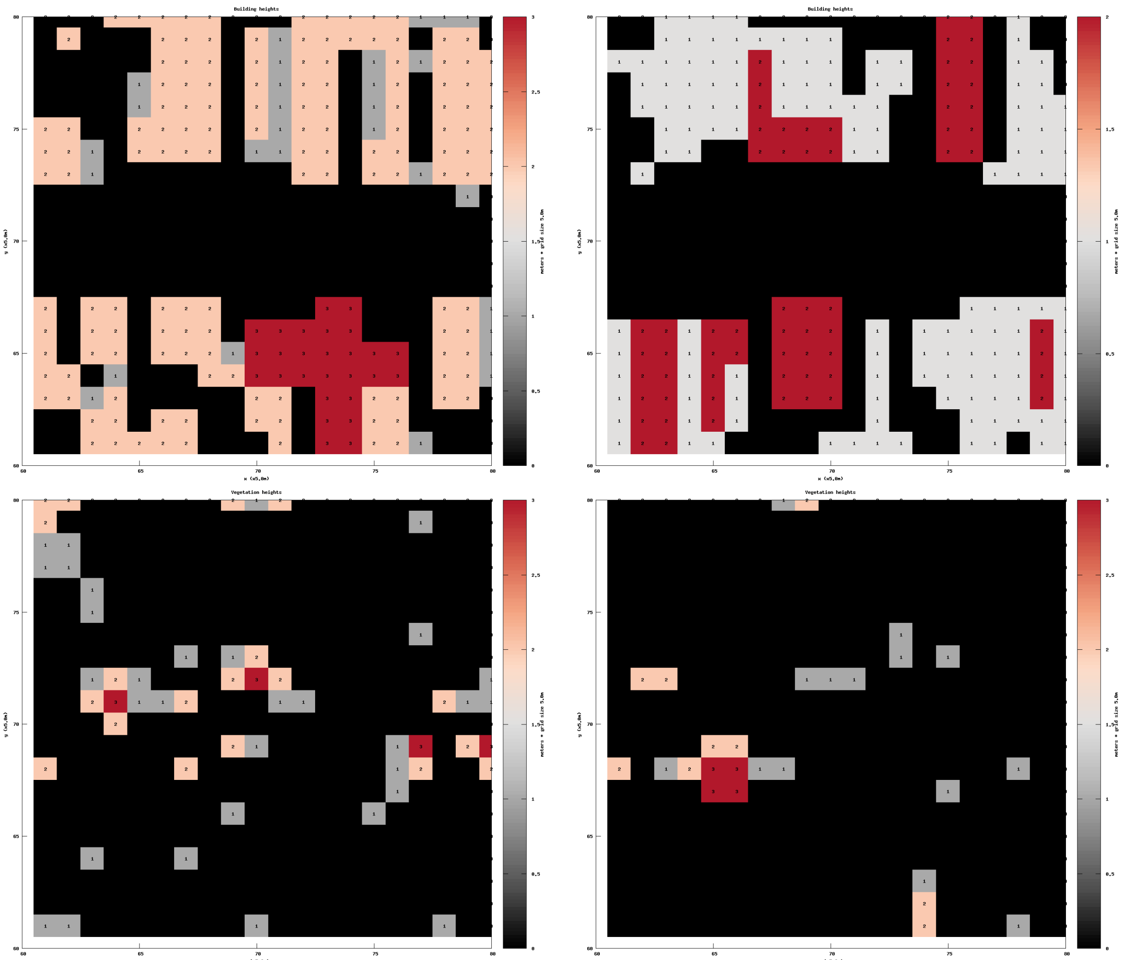

Figure 5.15: *Building heights (top) / Vegetation cover (bottom) - George St (left), Gipps St. (right)*

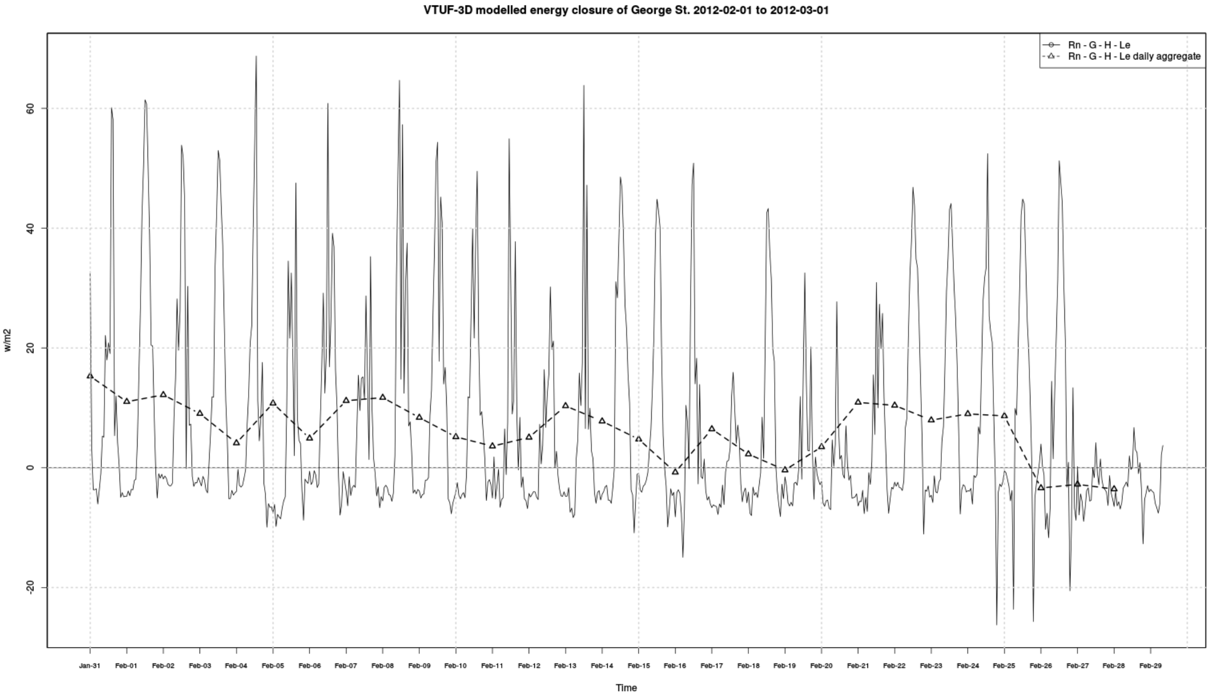

Figure 5.16: *Energy closure of George St*

calculated from the Tmrt, also reflecting those patterns. More work on improving the Tmrt calculations is another item on the future improvement list (Appendix .1.1). Tables 5.3, 5.4, and 5.5 summarize the RMSE and d index of agreement for Tmrt, Tsfc, and UTCI respectively (from Figures 5.18, 5.19, and 5.20). Tsfc values show a strong level of agreement, with d values ranging from 0.902 to 0.066, while Tmrt and UTCI show slightly less strong agreement with d values ranging from 0.804 to 0.896.

Table 5.3: *George St Tmrt predicted vs. observations*

Observation point	RMSE (°C)	d
EM12	12.6	0.846
EM9	13.9	0.804
EM3	11.6	0.896
EM2	15.0	0.832

Table 5.4: *George St Tsfc predicted vs. observations*

Observation point	RMSE (°C)	d
EM12	2.875	0.966
EM9	3.353	0.952
EM3	4.631	0.907
EM2	7.226	0.902

Table 5.5: *George St UTCI predicted vs. observations*

Observation point	RMSE (°C)	d
EM12	6.035	0.832
EM9	6.613	0.8
EM3	6.055	0.848
EM2	7.039	0.807

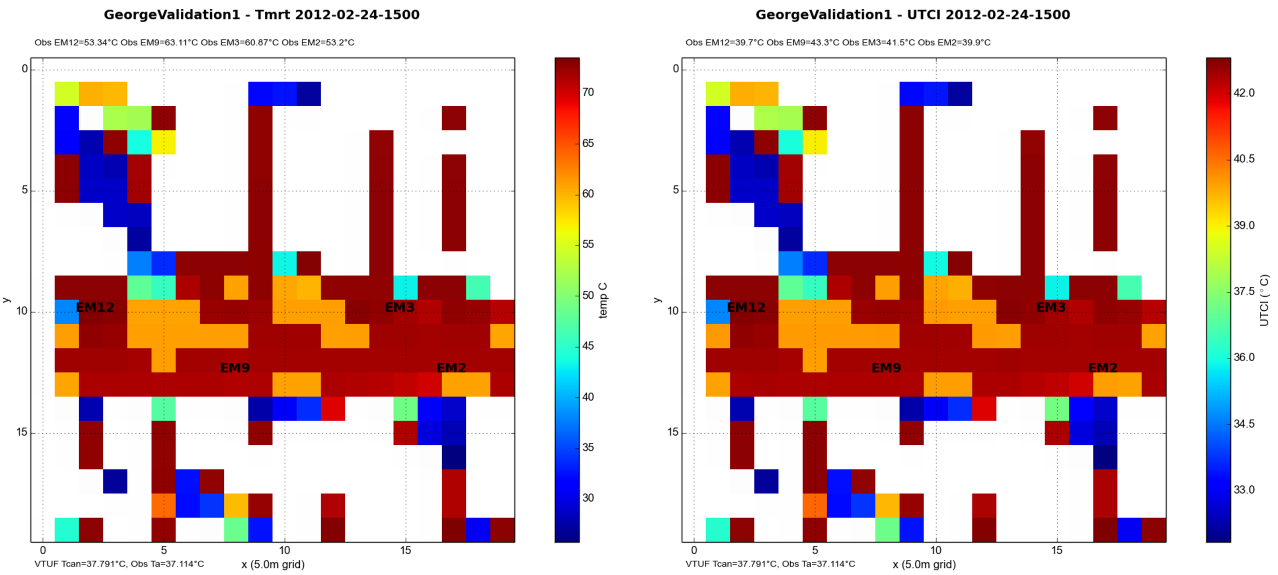

Figure 5.17: *Results of George St. Tmrt and UTCI for 24 February 2014 1500.*

5.1.5 Model testing and validation using City of Melbourne, Gipps St. dataset

In the Gipps St. validation, the energy balances (Figure 5.22) are much closer to full closure. Other analysis, predicted vs. three observed locations (Tables 5.6, 5.7, and 5.8, as well as Figures 5.24, 5.23, and 5.25) comparisons of Tmrt, Tsfc, and UTCI achieve results very similar to the George St. validations.

Table 5.6: *Gipps St. Tmrt predicted vs. observations*

Observation point	RMSE (°C)	d
EM5	5.892	0.915
EM11	4.463	0.931
EM8	6.086	0.906

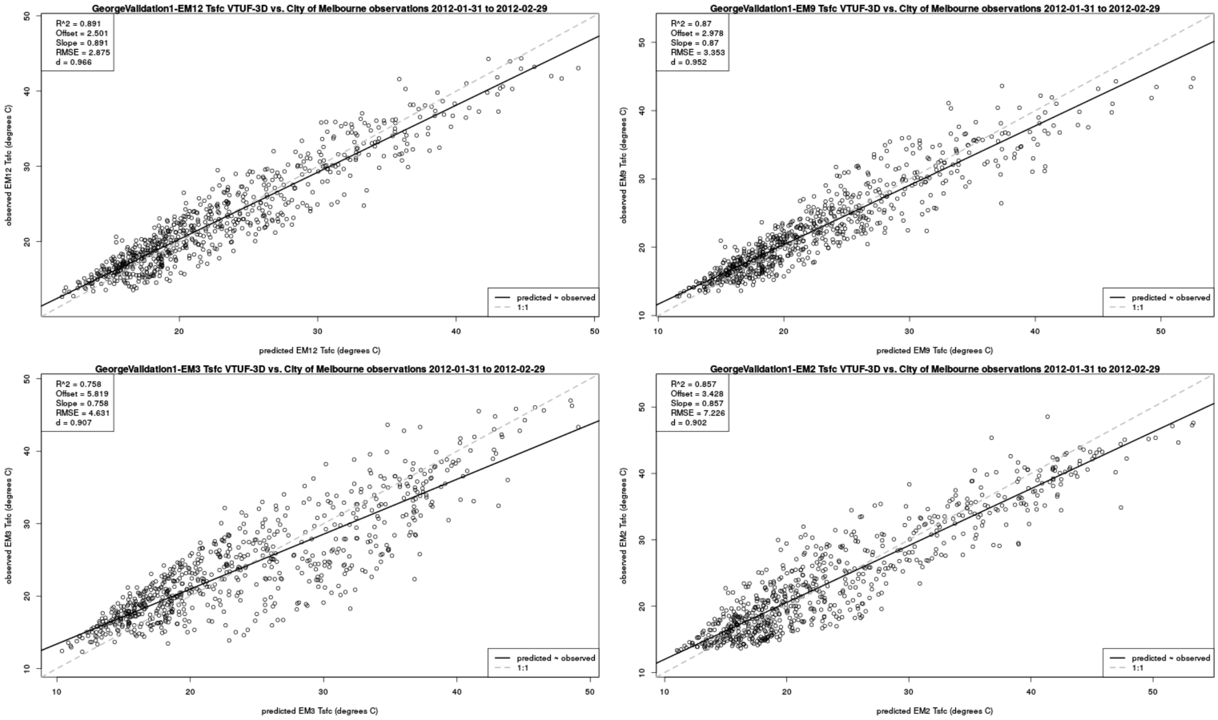

Figure 5.18: *George St. point comparison of Tsfc of 4 observation stations to modelled points*

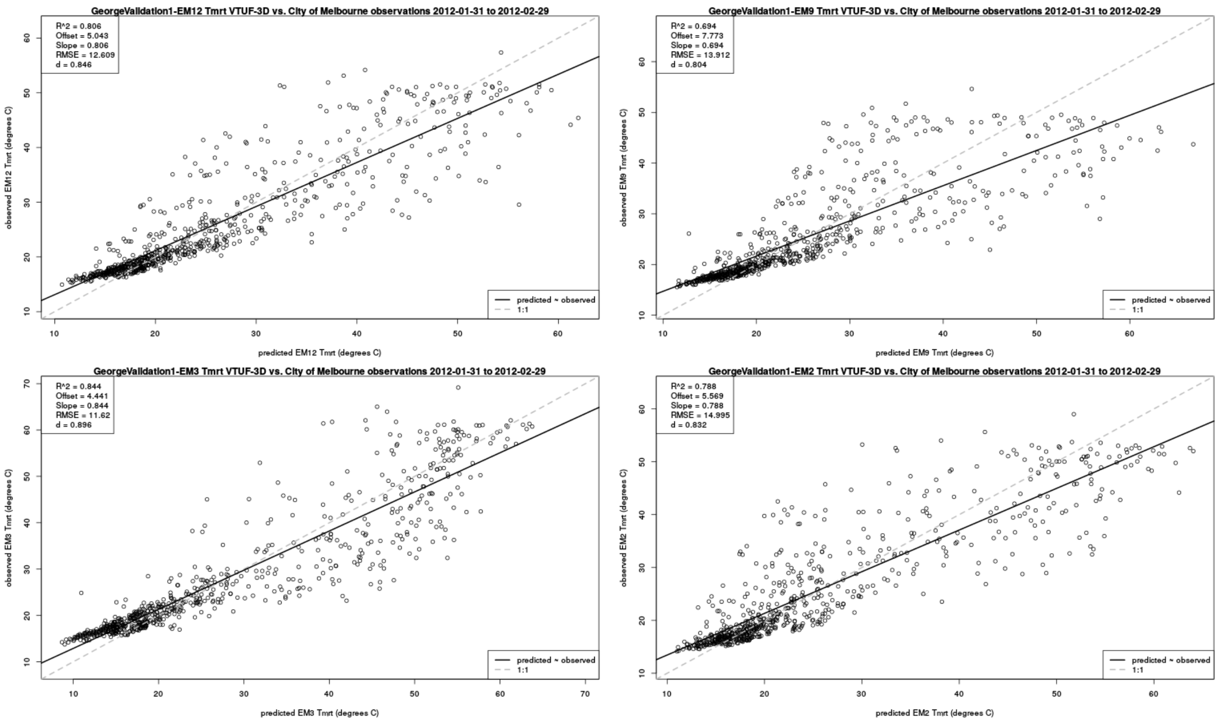

Figure 5.19: *George St. point comparison of Tmrt of 4 observation stations to modelled points*

Table 5.7: *Gipps St. Tsfc predicted vs. observations*

Observation point	RMSE (°C)	d
EM5	12.398	0.896
EM11	12.959	0.865
EM8	12.781	0.889

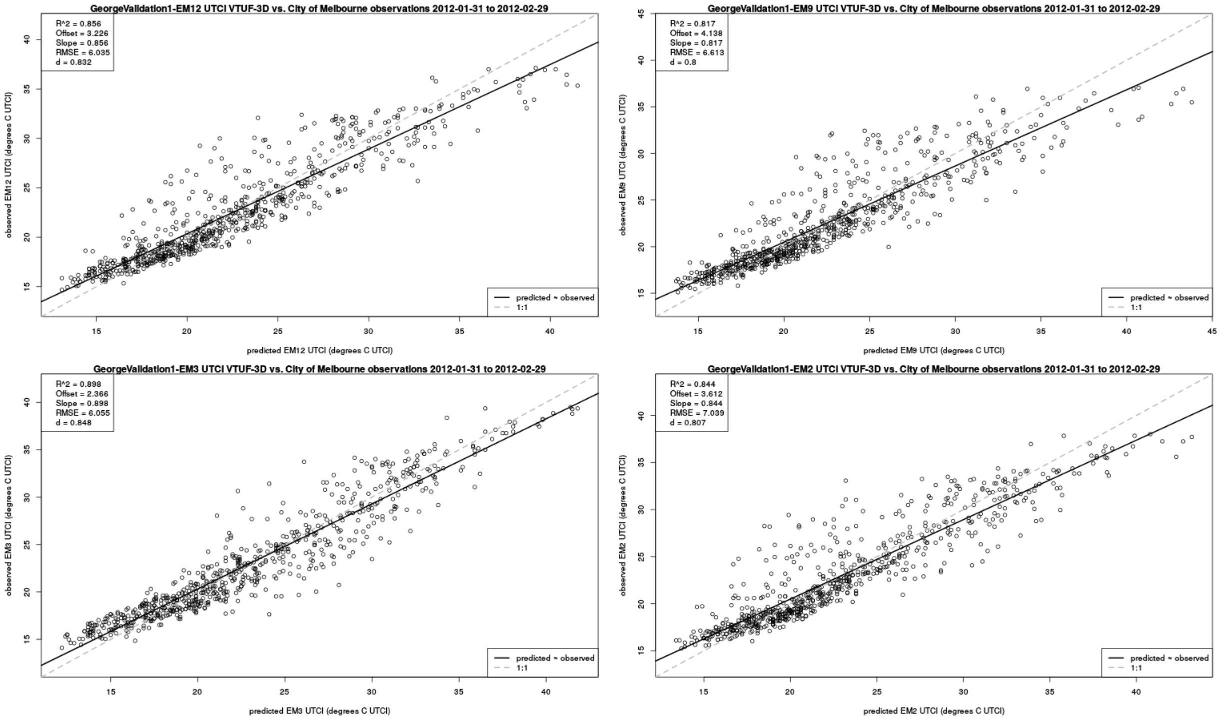

Figure 5.20: *George St. point comparison of UTCI of 4 observation stations to modelled points*

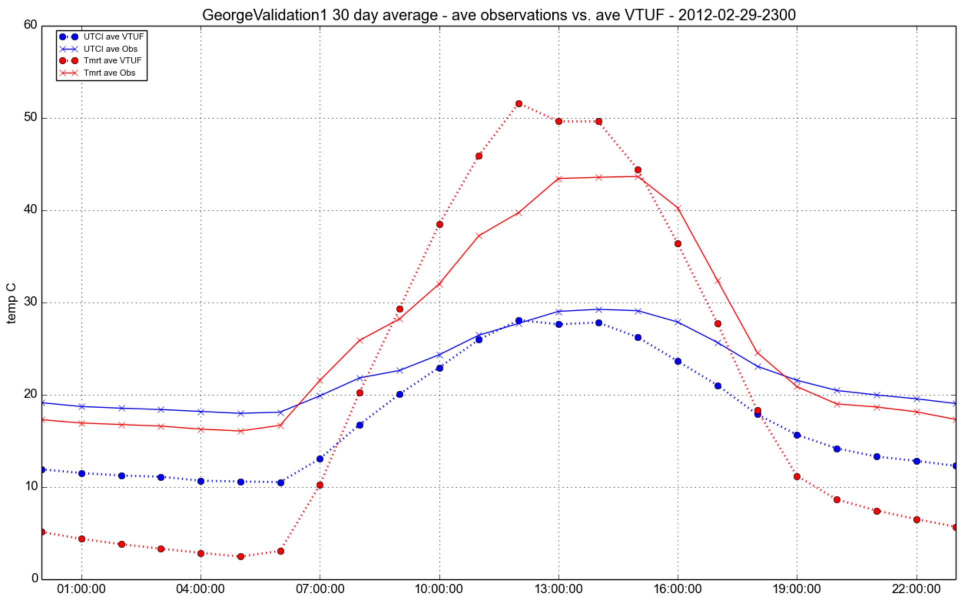

Figure 5.21: *George St. 30 day averages, 4 observation stations compared to modelled points*

Table 5.8: *Gipps St. UTCI predicted vs. observations*

Observation point	RMSE (°C)	d
EM5	5.773	0.874
EM11	6.289	0.834
EM8	6.085	0.854

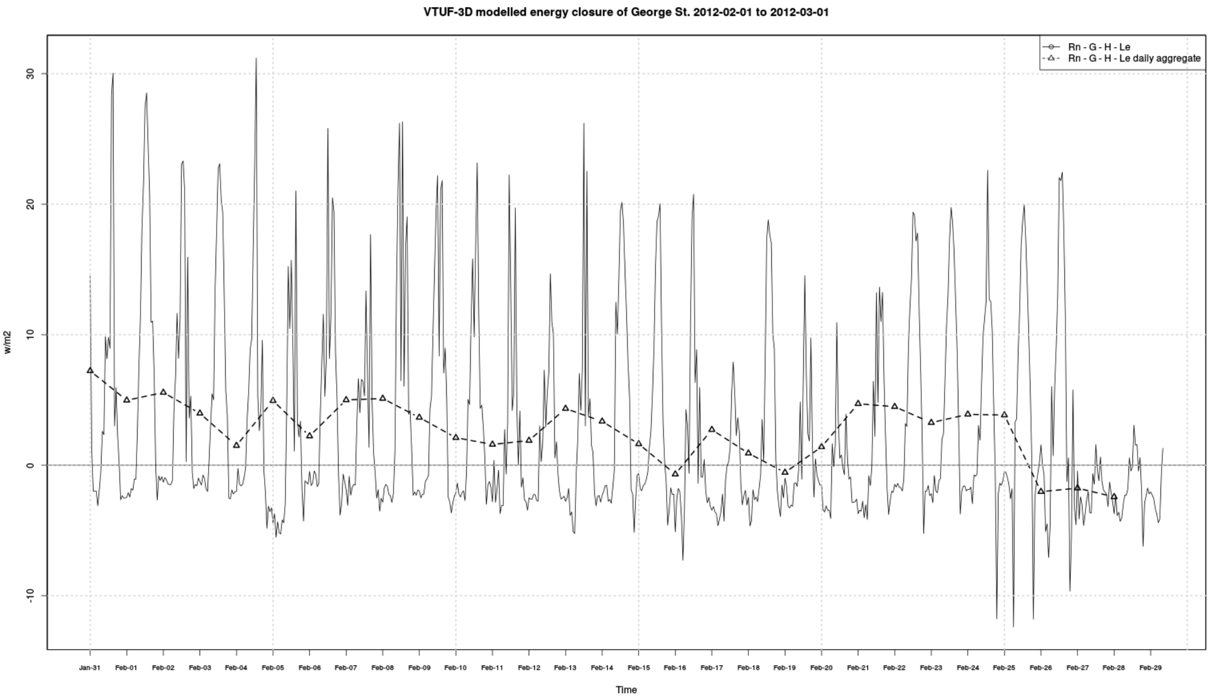

Figure 5.22: *Energy closure of VTUF-3D and observations for Gipps St.*

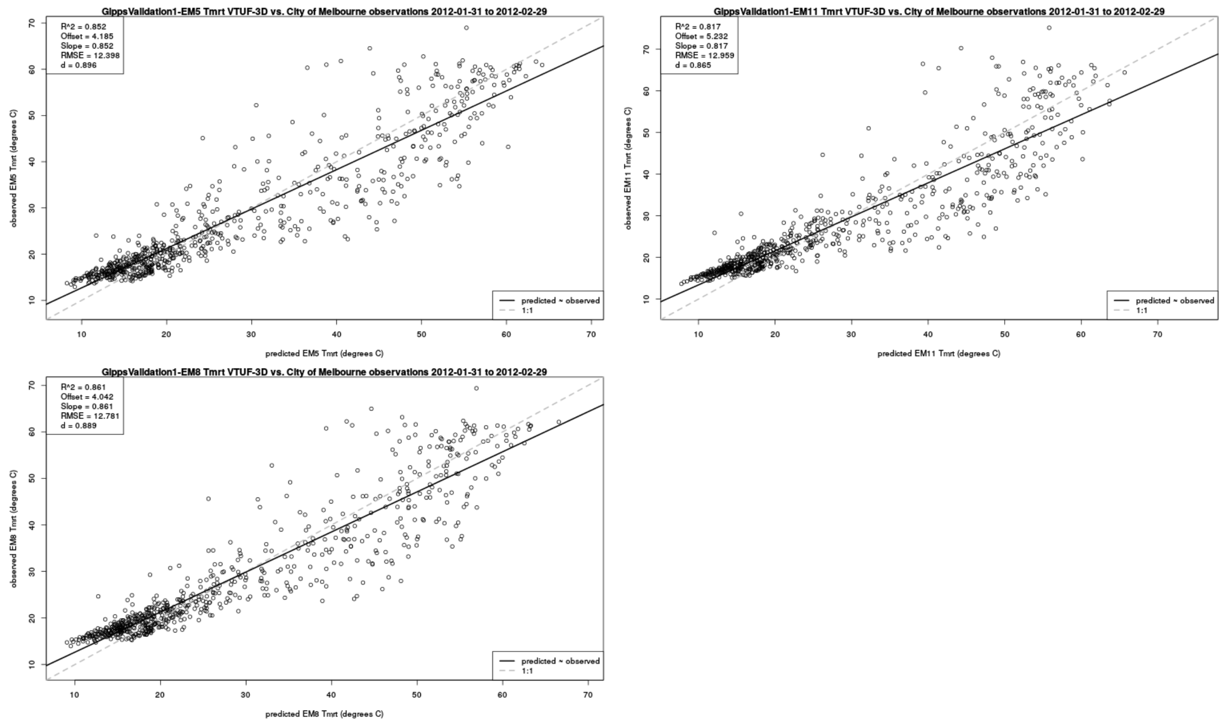

Figure 5.23: *Gipps St. point comparison of Tmrt of 3 observation stations to modelled points*

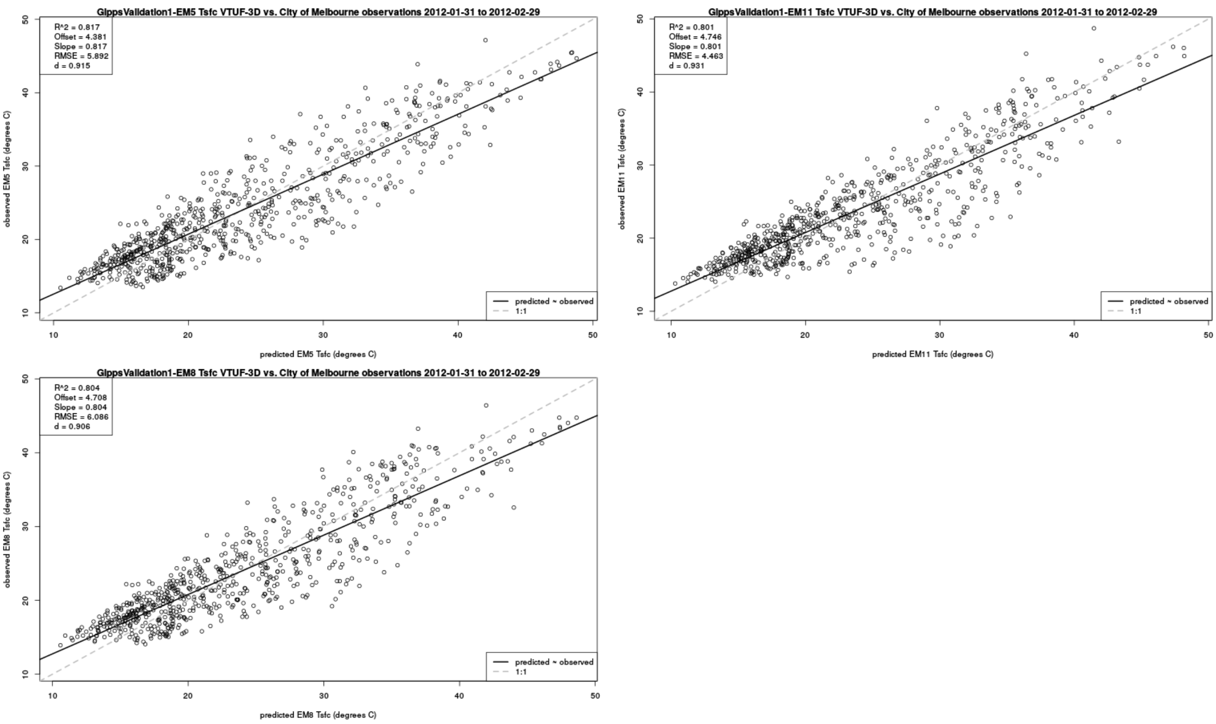

Figure 5.24: *Gipps St. point comparison of Tsfc of 3 observation stations to modelled points*

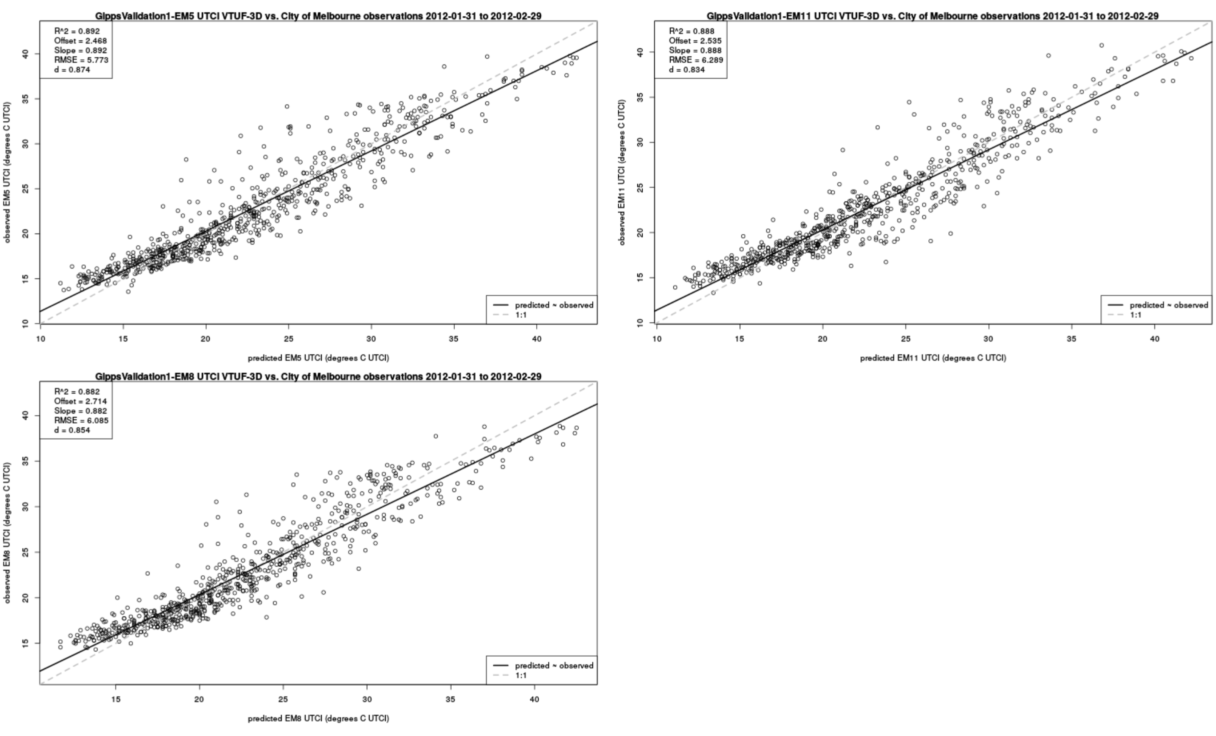

Figure 5.25: *Gipps St. point comparison of UTCI of 3 observation stations to modelled points*

5.1.6 Model testing and validation using Lincoln Sq dataset

A third set of validations was run against the Lincoln Square data set (Motazedian 2015). This is a Melbourne urban square (Figure 5.26) with a mix of open grass and mature trees (Figure 5.28) within a dense urban canyon (Figure 5.27). Domain resolution was 10m grids.

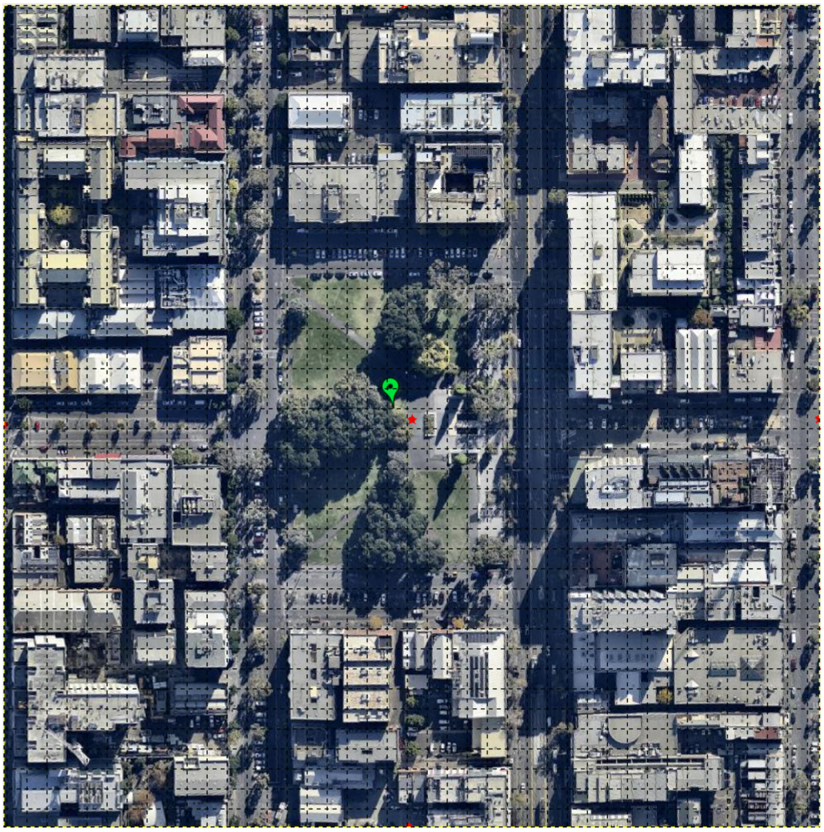

Figure 5.26: *Lincoln Square domain*

The modelled results were compared to transits of Tsfc (Figure 5.29) and UTCI (Figure 5.30) and showed broad agreement with the spatial patterns in the observations.

5.2 Future tasks

5.2.1 Model testing and validation using Hughesdale dataset

Future validations are to be done against the Hughesdale data set. These observations are currently being recorded and finalized. Hughesdale (Figure 5.31) is a medium density homogeneous suburb in the south east of Melbourne, Australia. These observations will include evapotranspiration and other tree physiology observations in addition to climate (air temperature, humidity, shortwave radiation, and wind speed) observations.

5.2.2 Model testing and validation using Smith St dataset

A final set of validations are to be done against the Smith Street data set (Gebert et al. 2012). This is an inner suburb of Melbourne, Australia. This street is a higher density/retail area, with a high percentage of impervious surfaces and very sparse tree cover. Observations were done

Figure 5.27: *Lincoln Square building heights*

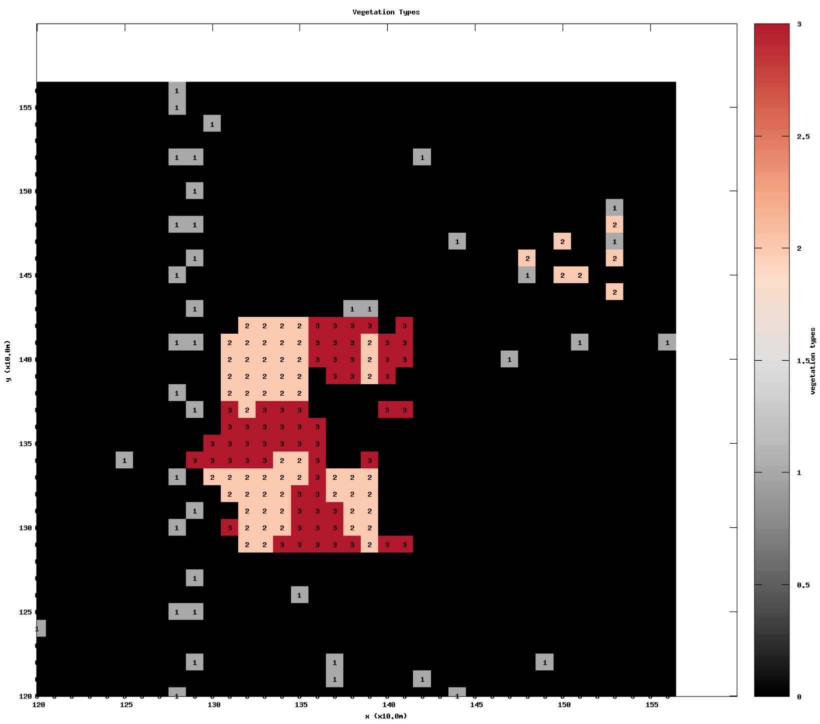

Figure 5.28: *Lincoln Square vegetation heights*

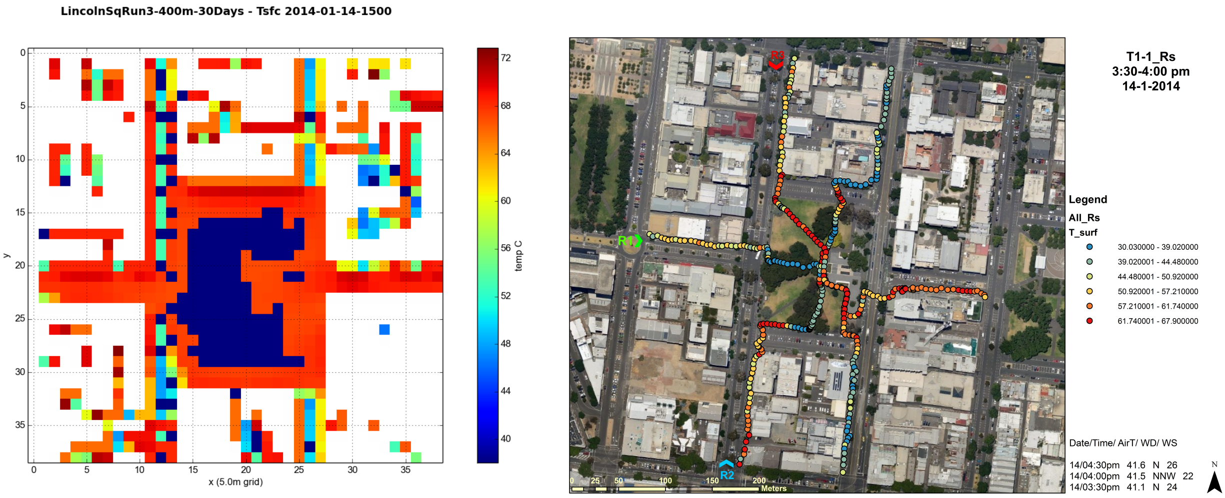

Figure 5.29: *Comparisons of modelled Tsfc to observed transits of Lincoln Sq. on 14 January 2014 3pm*

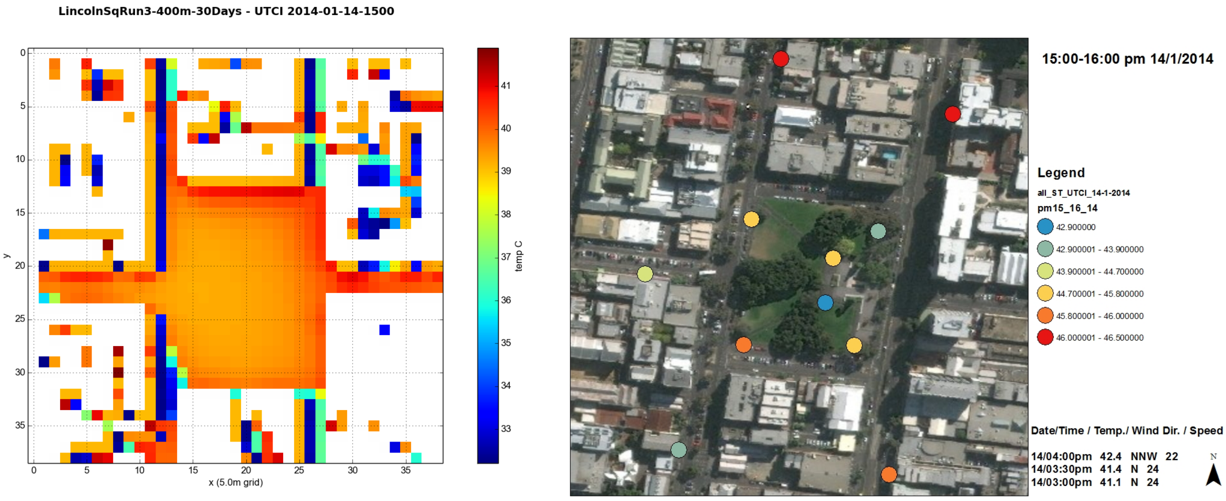

Figure 5.30: *Comparisons of modelled UTCI to observed transits of Lincoln Sq. on 14 January 2014 3pm*

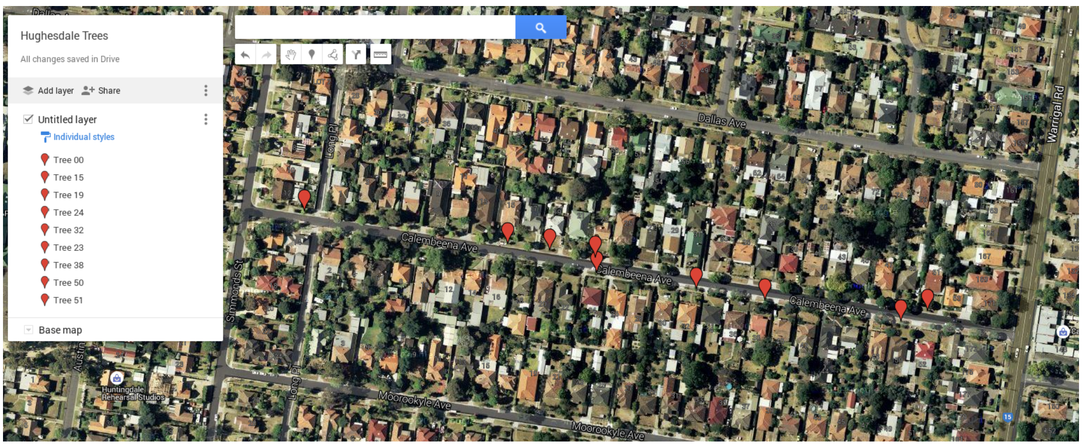

Figure 5.31: *Hughesdale suburb and observation station locations*

of physiological processes of isolated trees in addition to the regular climate parameters. These validations will compare modelled values of evapotranspiration and photosynthesis to observed values.

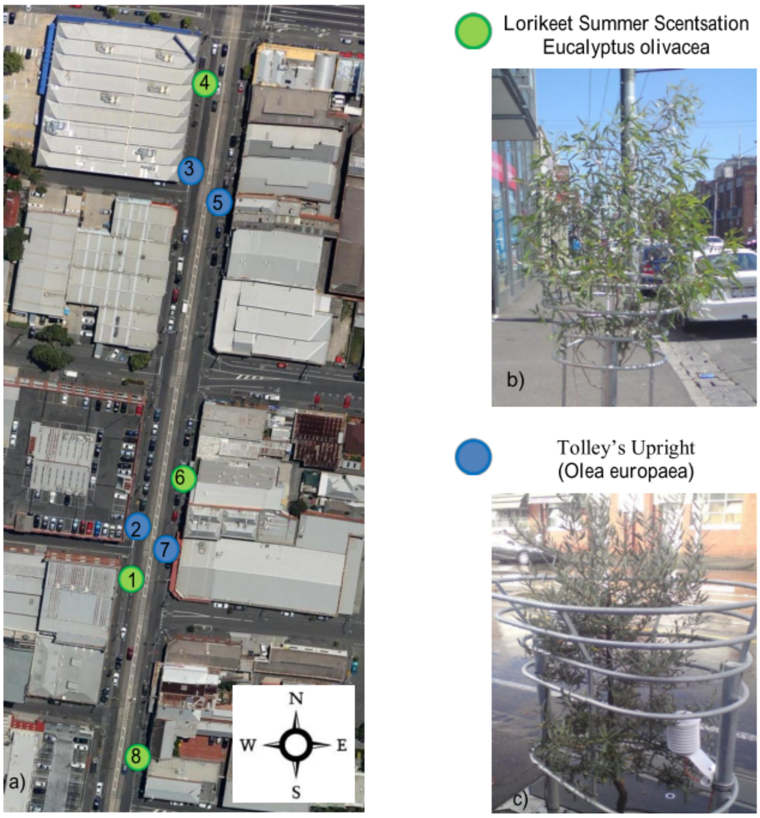

Figure 5.32: *Smith St. contains detailed physiological observations of isolated trees*