Overview

Tristan Dyer

In this assignment, we will be updating the 2013 LIDAR DEM with data from a 2015 UAV derived point cloud. We'll then perform a water flow simulation and animate and discuss the results. The idea here is to understand the process of preparing data and performing the water flow simulation so that multiple simulations can be performed in order to show how differences in terrain can effect the flow and resulting accumulation of water.

Smooth Fusion

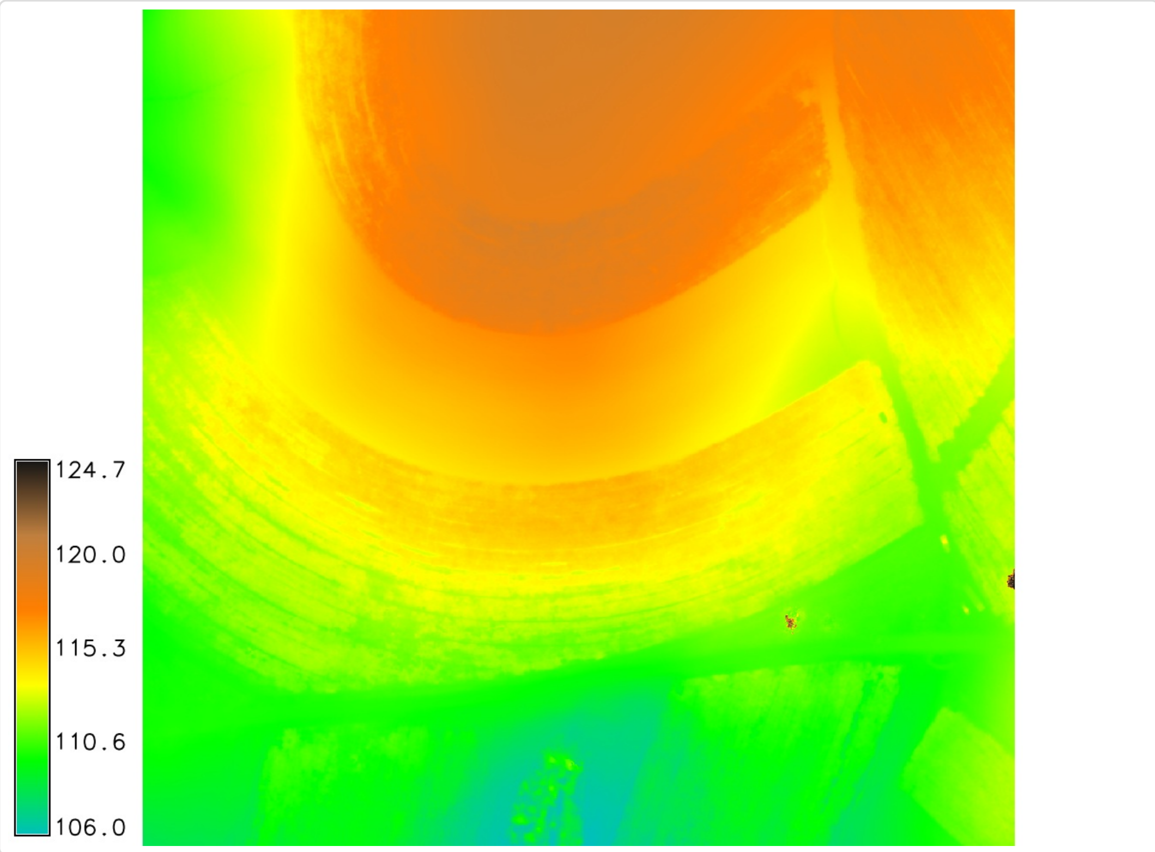
First, we'll do a smooth fusion of LIDAR and UAS data in order to update the 2013 DEM to the 2015 state of the terrain. Set the region and import the UAS derived point cloud data:

```
g.region n=219657 s=219320 e=637082 w=636731 res=0.3 -p
v.in.lidar -trbo input=2015_06_20_Points_11GCP_Ncspm.las
output=uav_06_all
```

Note the **t, b,** and **r** flags with the `v.in.lidar` command. Importing point cloud data can be time consuming, but these flags should help speed things up by telling GRASS not to create an attribute table, not to build topology, and to only import points that are inside of the current region. Next, we'll interpolate the LIDAR data to create a raster surface:

```
v.surf.rst npmin=100 input=uav_06_all elevation=uav_06_interp
tension=30 smooth=1
```

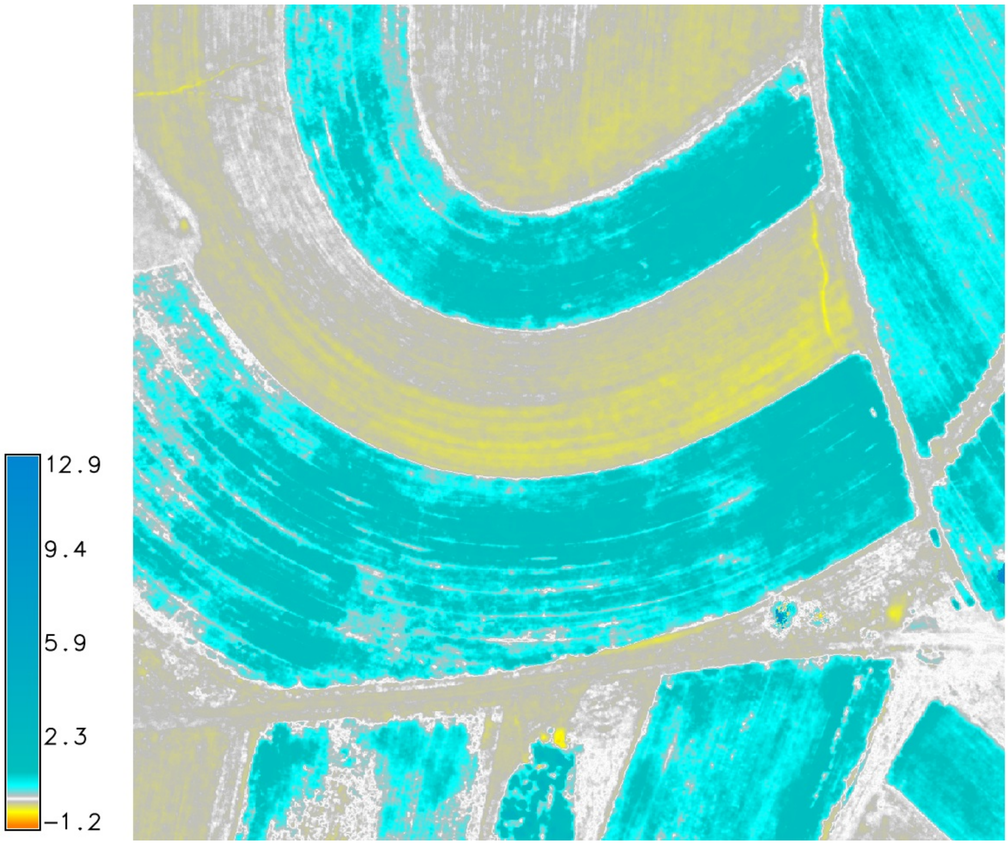
The `v.surf.rst` command generates a raster surface from vector data using a regularized spline with tension. The **npmin** option defines the minimum number of points needed to approximate a segment. The **tension** and **smooth** parameters are the tension and smoothing values used in the spline. This generates the following DSM:



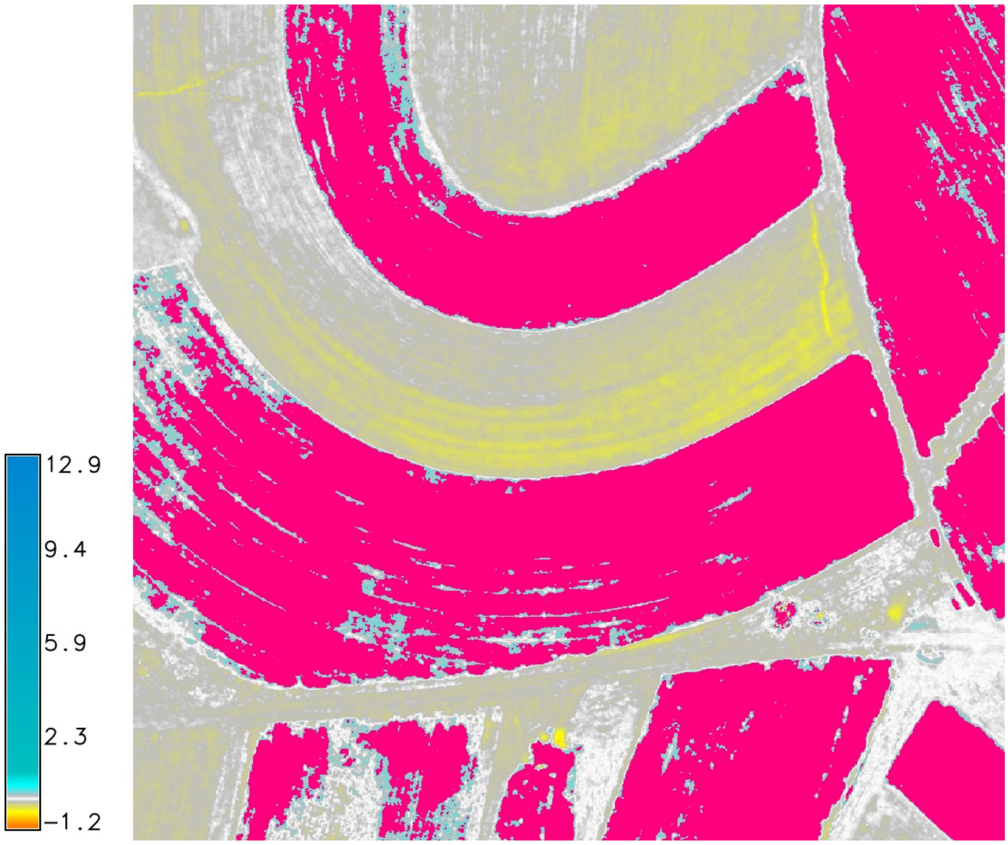
Interpolated UAS Surface

Next, we want to mask out any vegetation in the interpolated DSM so that it won't be included when we perform the fusion with the 2013 DEM. In order to create the mask, we'll assume that anything in the dataset that is more than 30cm higher than the 2013 dataset is vegetation. This is a fairly good assumption, as we'd mostly like to mask out crops (corn, which grows to larger than 30cm tall), trees, and buildings, and we're fairly certain that no natural processes have caused over 30cm of soil accretion in our study area. Perform the subtraction and create the mask:

```
r.mapcalc "diff_uav_06_lidar = uav_06_interp -
mid_pines_lidar2013_dem"
r.mapcalc "diff_uav_06_lidar_mask = if (uav_06_interp -
mid_pines_lidar2013_dem > 0.3, 1, null())"
```



Interpolated UAS surface minus the 2013 LIDAR DEM



Masked Vegetation

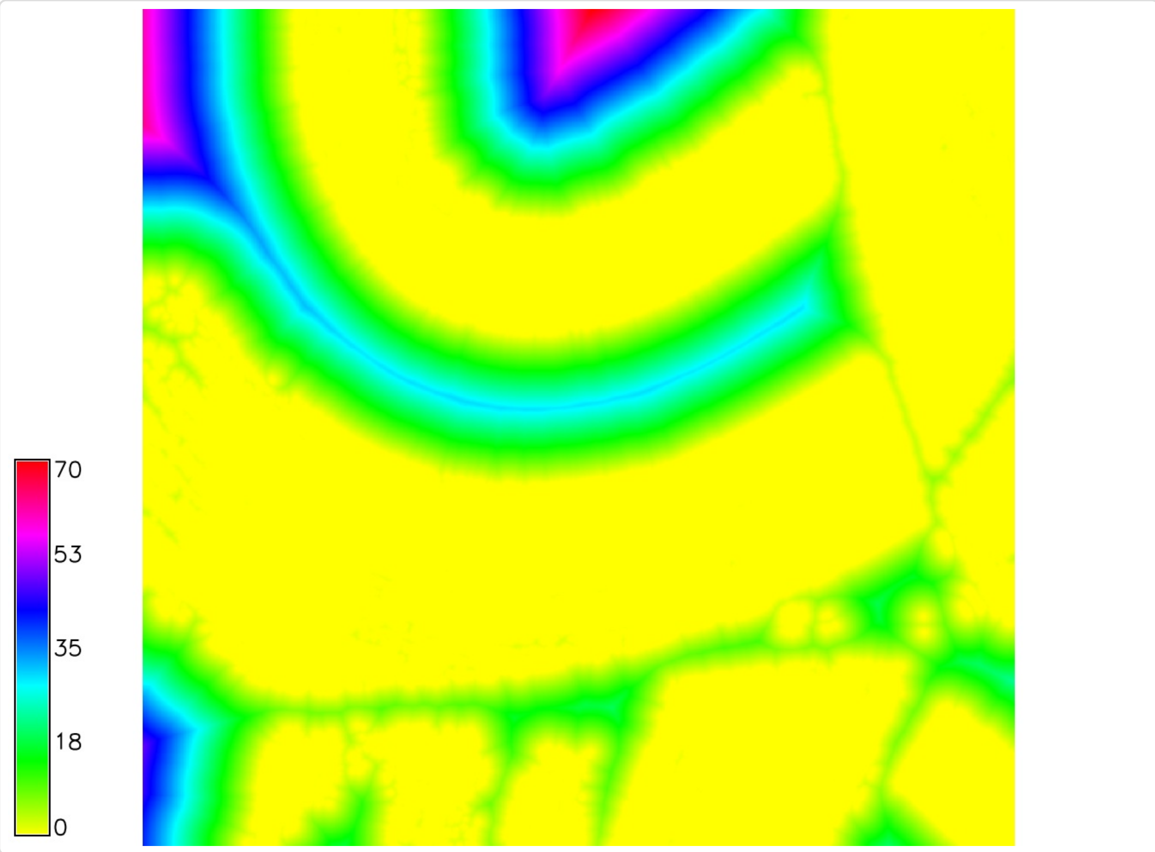
We can see that the mask does a fairly good job of covering the crops, trees, and buildings. In order to fill some of the holes in the mask, we'll dilate it using `r.grow`

```
r.grow input=diff_uav_06_lidar_mask
output=diff_uav_06_lidar_mask_grow radius=3.01 new=1
```

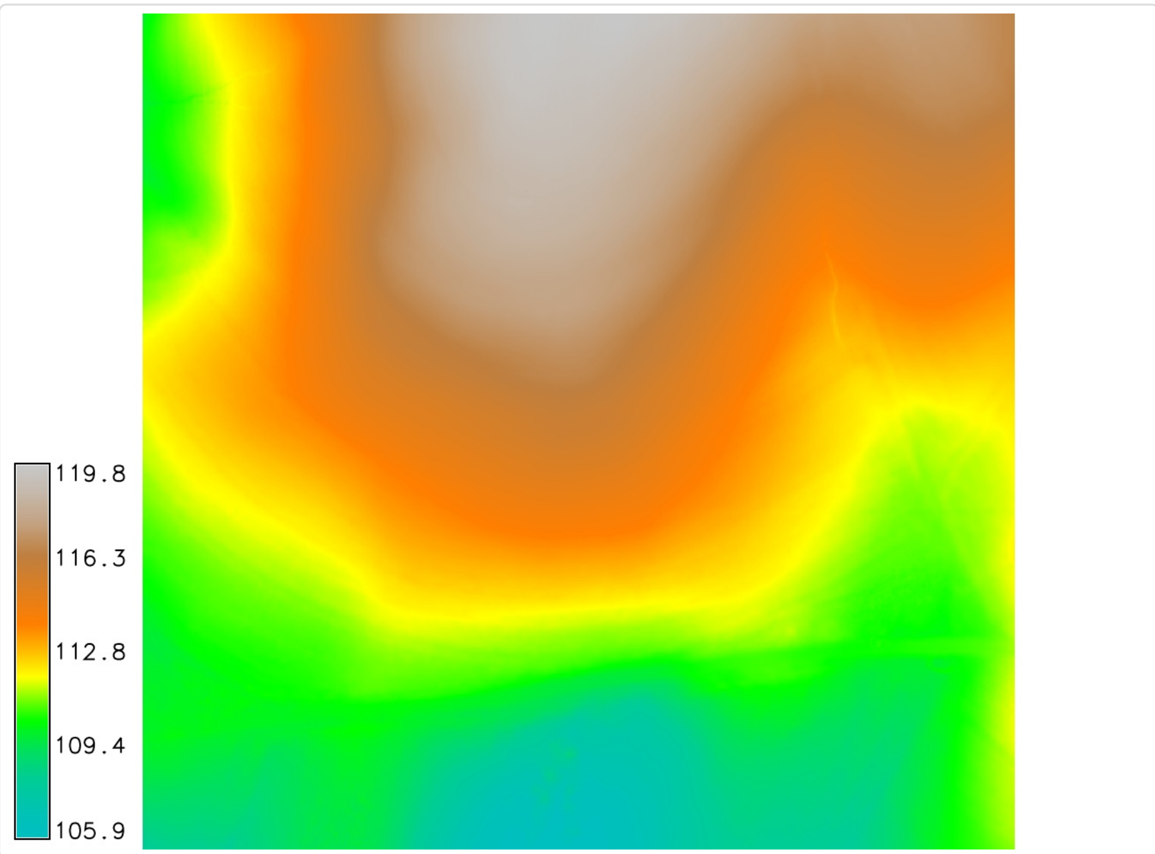
The dilated vegetation mask

This mask represents the area which will be replaced with data from the 2013 LIDAR DEM. To perform the data fusion, we'll use a weighted average method, similar to what we did in Assignment 6. For each cell, we calculate the shortest distance from the mask. If the distance is greater than 10 cells, we keep the value from the UAS interpolated surface. If the distance is within 10 cells, we take the weighted average of the UAS and LIDAR data, using distance from the masked cells as the weights.

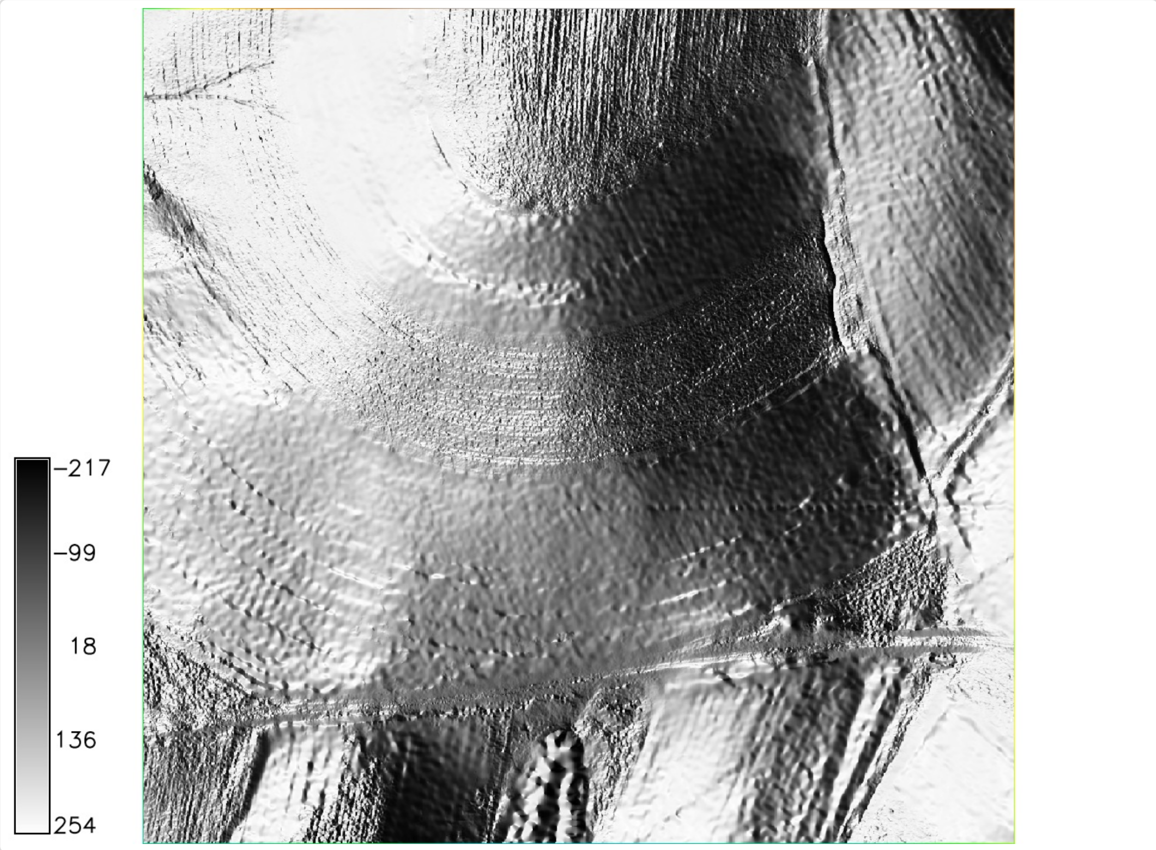
```
r.grow.distance -m input=diff_uav_06_lidar_mask_grow
distance=distance_mask
r.mapcalc merged_lidar_uav = "if(distance_mask > 10, uav_06_interp,
(1 - distance_mask/10) * mid_pines_lidar2013_dem + (distance_mask/10)
* uav_06_interp)"
r.relief input=merged_lidar_uav output=merged_lidar_uav_relief
zscale=100
```



The distance mask



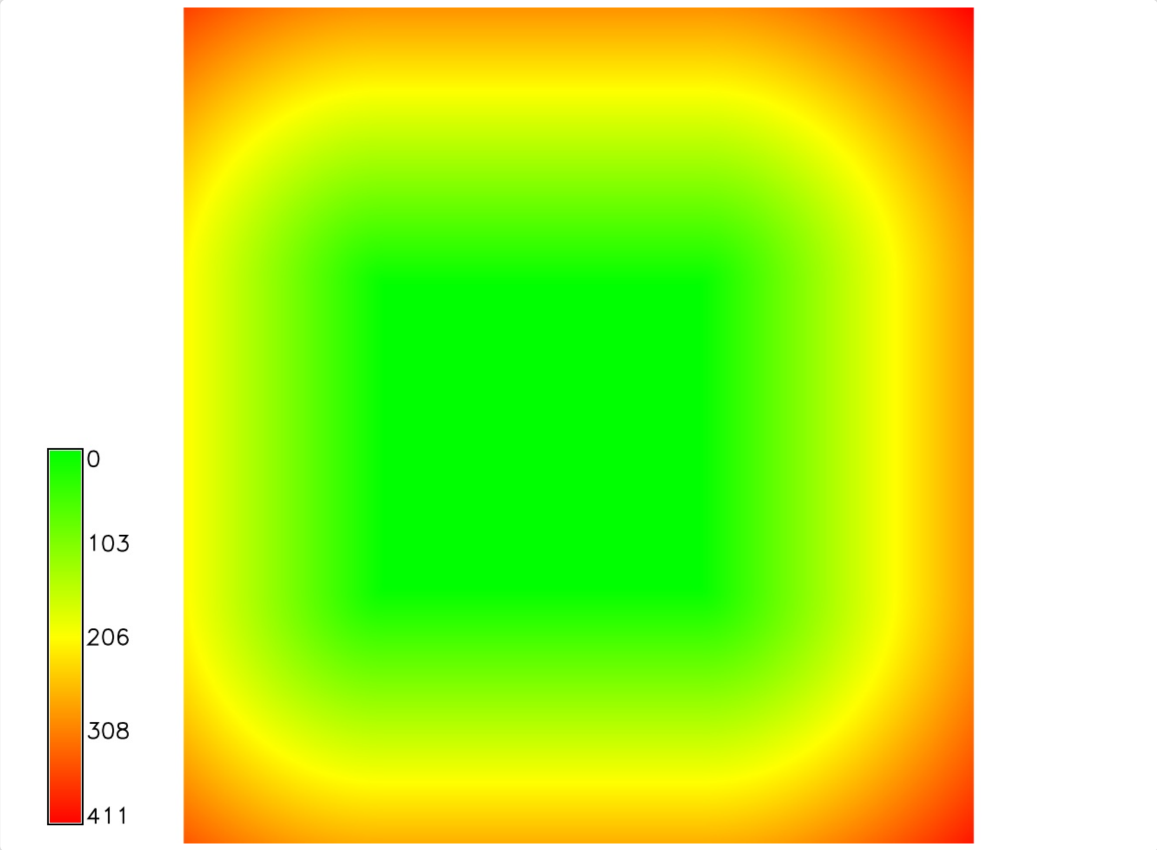
The merged dataset



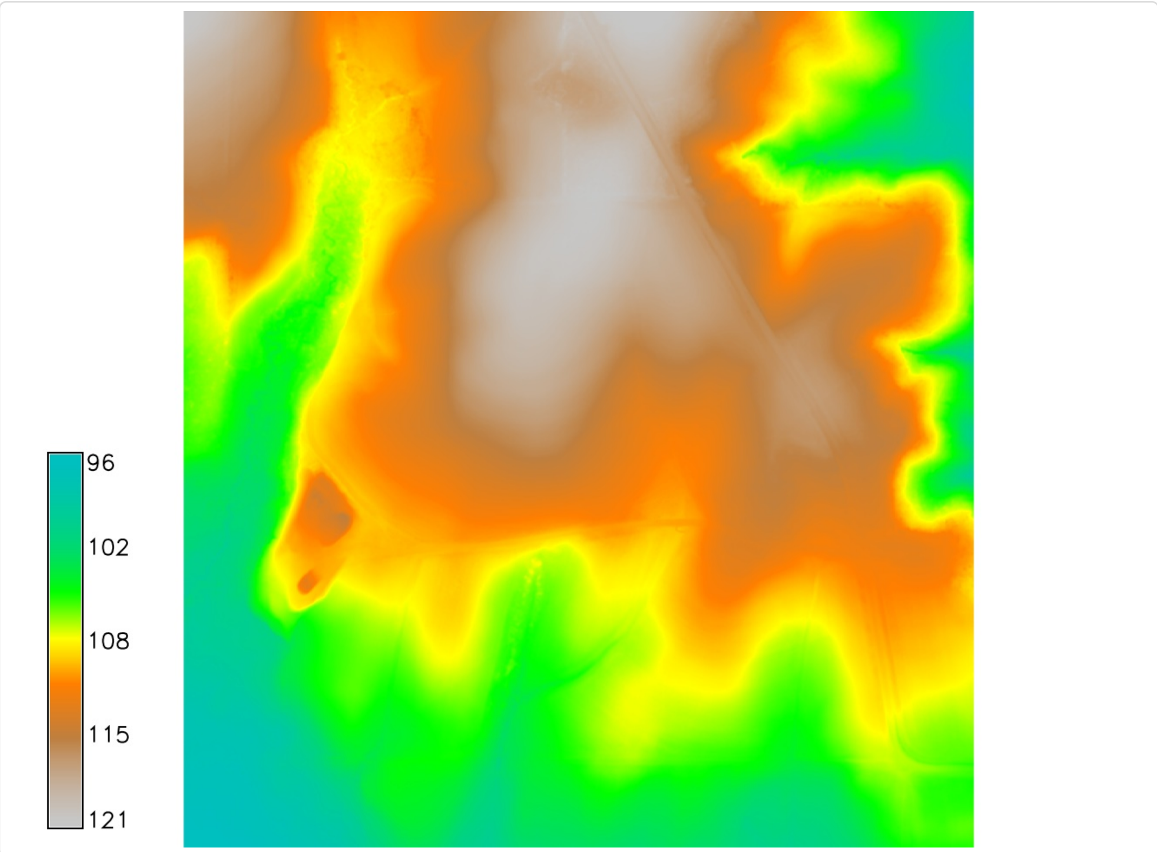
Relief of the merged dataset

In the first image above, we see the distance mask, which contains the distance from each cell to the vegetation mask. The second image shows the merged dataset, and the third shows a relief of the merged dataset. From the relief we can see the smooth transition that has been created between areas of the new UAS derived dataset and areas that were filled using the older 2013 LIDAR dataset. We can also see that the 2013 dataset has quite a bit more smoothing applied, which is important to note as we move into performing water flow simulations. Finally, we'll merge the combined dataset into the larger area covered by the LIDAR dataset using the same technique:

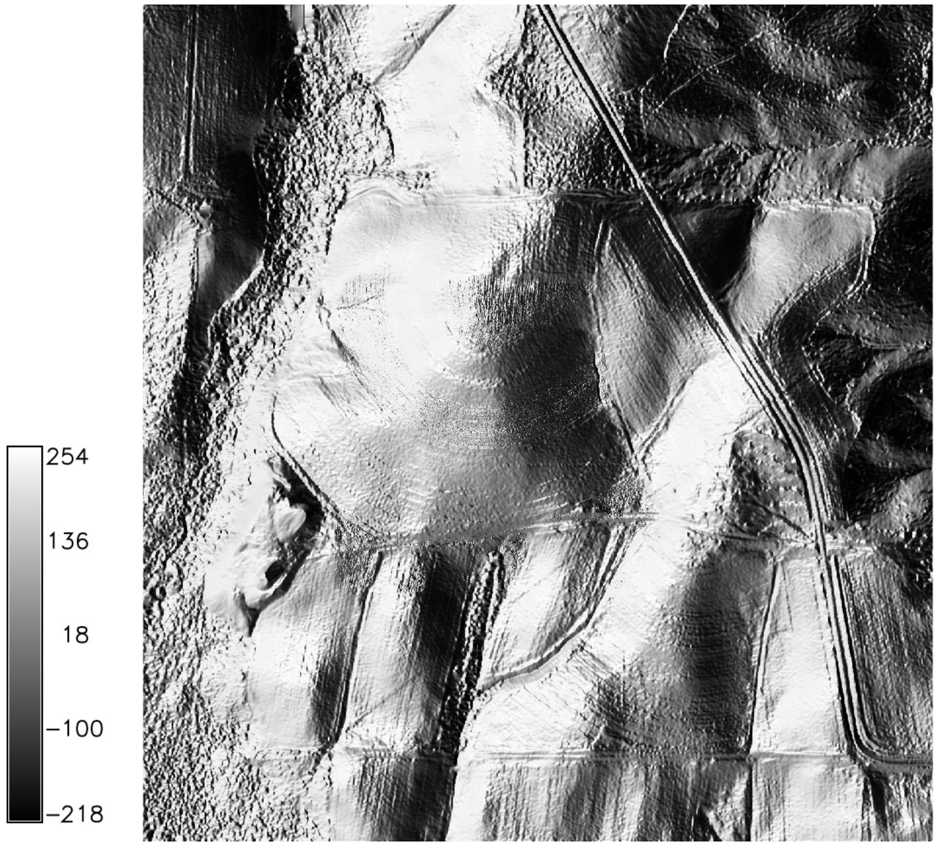
```
g.region n=n-10 s=s+10 e=e-10 w=w+10
r.mapcalc "rectangle = 1"
g.region raster=mid_pines_lidar2013_dem
r.grow.distance -m input=rectangle distance=distance_rectangle
r.mapcalc "merged_lidar_uav_full = if(distance_rectangle > 10,
mid_pines_lidar2013_dem, (1 - distance_rectangle/10) *
merged_lidar_uav + (distance_rectangle/10) *
mid_pines_lidar2013_dem)"
r.relief input=merged_lidar_uav_full
output=merged_lidar_uav_full_relief zscale=100
```



The distance mask



The merged dataset



Relief of the merged dataset

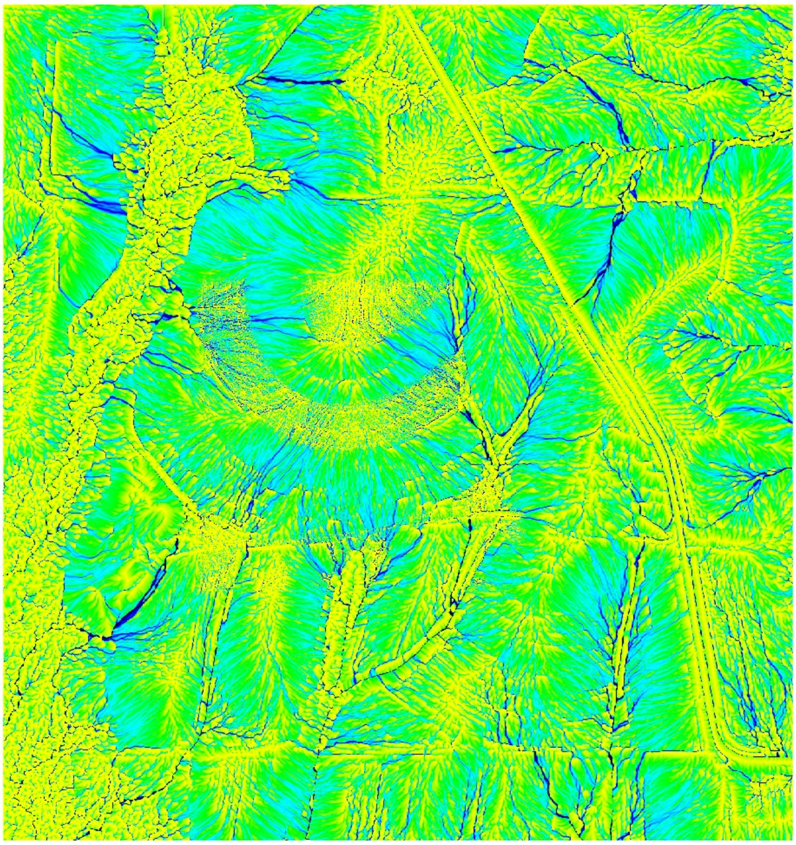
Again, the first image shows the distance from each cell to the mask (which is a large rectangle covering the entire merged dataset). The second image shows the final merged dataset, and the third shows a relief of the final merged dataset. Inspecting the relief image (open image in new tab for larger view) will show the smooth transition between the inner UAV data with the outer LIDAR data.

Water Flow Simulation

In this section, we'll perform a water flow simulation in order to inspect the flow patterns that are created using different DSMs. This workflow shows how to perform the simulation for the merged 2013 LIDAR/2015 UAS dataset we just created, but the same process should be executed on the timeseries of UAS datasets (March, July, September, and October) to show how the flow patterns evolve. First, examine the watershed boundaries to see which areas to include:

```
r.watershed elevation=merged_lidar_uav_full threshold=50000
accumulation=merged_flowacc basin=merged_basin
```

Merged dataset watershed basins



Merged dataset flow accumulation

We'll use a subset of this large region to perform the water flow simulation. We'll use a region that include the entire merged dataset so that we can look for artifacts introduced by the merging process. Additionally, we'll need first order spatial derivatives to perform the water flow simulation. These can be generated using `r.slope.aspect`.

```
g.region n=219727 s=219319 e=637191 w=636701 res=0.3 -p
r.slope.aspect elevation=merged_lidar_uav_full dx=merged_dx
dy=merged_dy
```

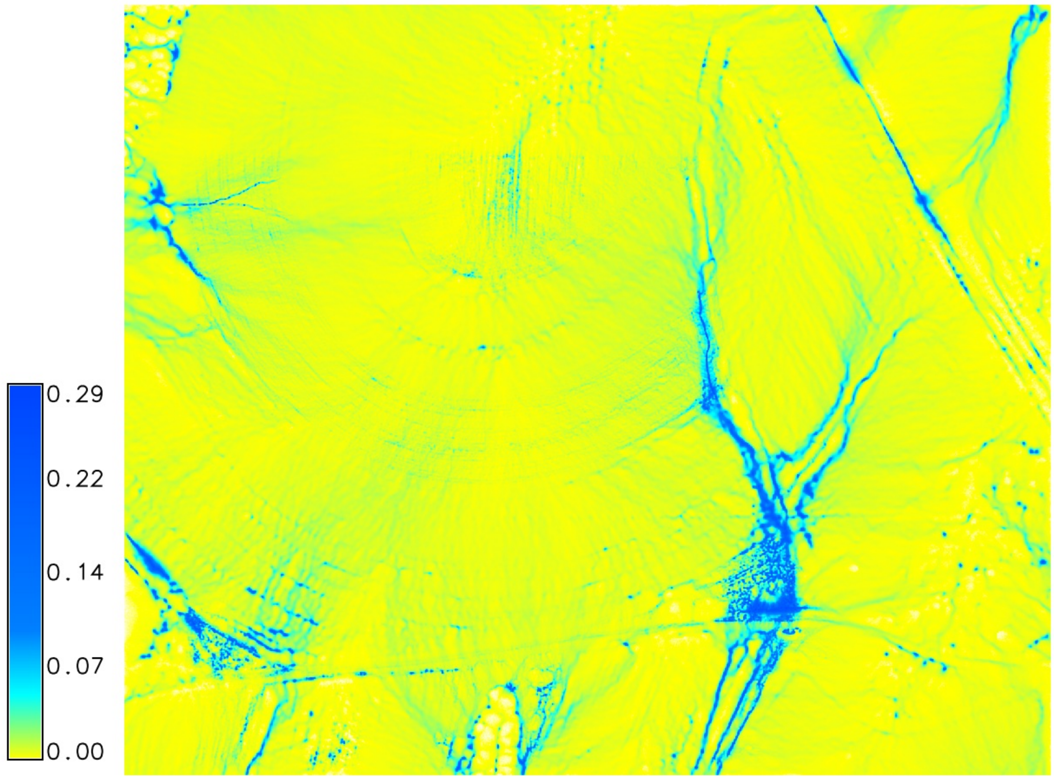
Finally, perform the water flow simulation:

```
r.sim.water -t elevation=merged_lidar_uav_full dx=merged_dx
dy=merged_dy rain_value=30 man_value=0.15 depth=merged_depth
discharge=merged_disch nwalkers=100000 niterations=20 output_step=1
hmax=0.2 halpha=8.0 hbeta=1.0
```

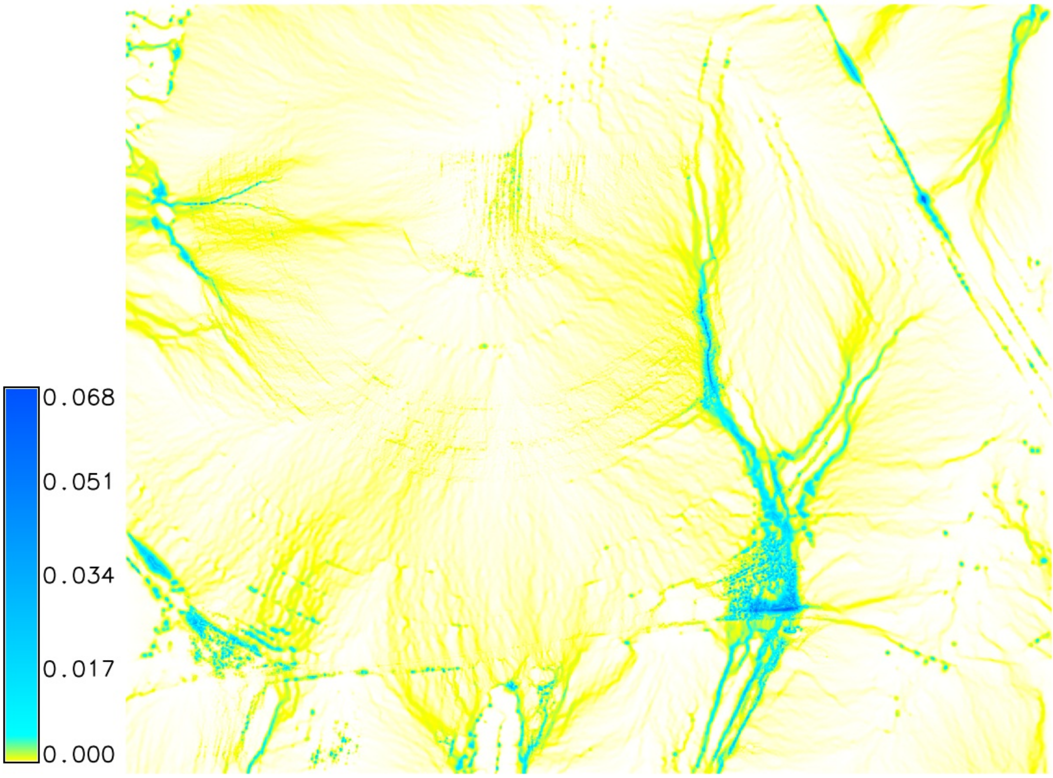
This simulation is fairly time consuming, and can be sped up by using a smaller region. A percentage completion should help you estimate how long the process will take. My simulation resulted in the following output:

```
Min elevation = 105.85 m
Max elevation = 120.42 m
Mean Source Rate (rainf. excess or sediment) = 0.000006 m/s or kg/m2s
Mean flow velocity = 1.474124 m/s
Mean Mannings = 0.150110
Number of iterations = 5895 cells
Time step = 0.05 s
100%
```

The following two images show the final timestep of the simulation:



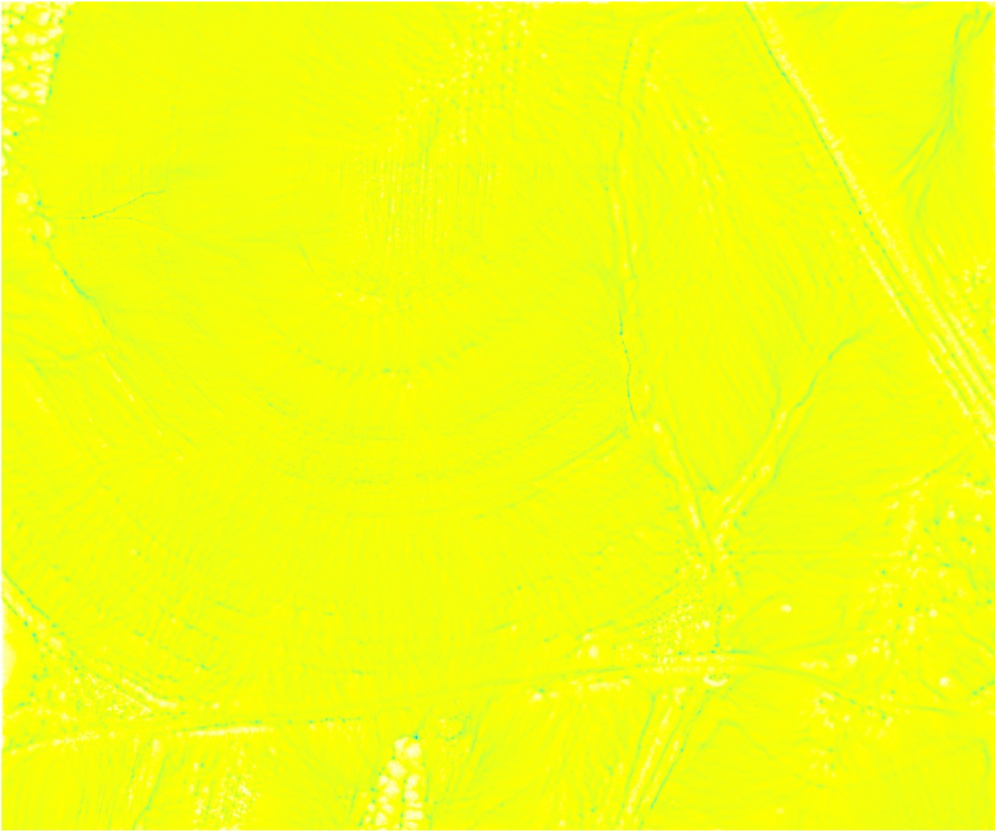
Depth - Final Timestep



Discharge - Final Timestep

Animation and Visualization

Using the Animation Tool in GRASS GIS, we can visualize how the water flow evolves over time, as seen in the following two animated gifs:



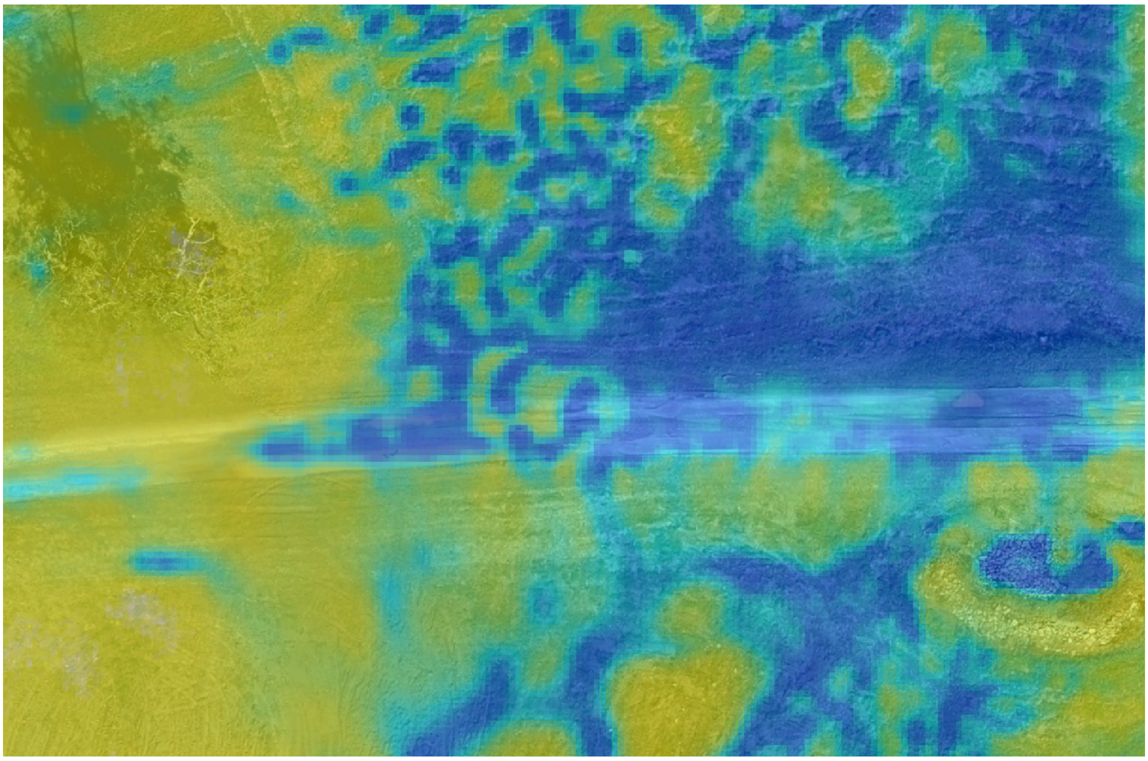
Depth

Discharge

Validating Using Orthophotos

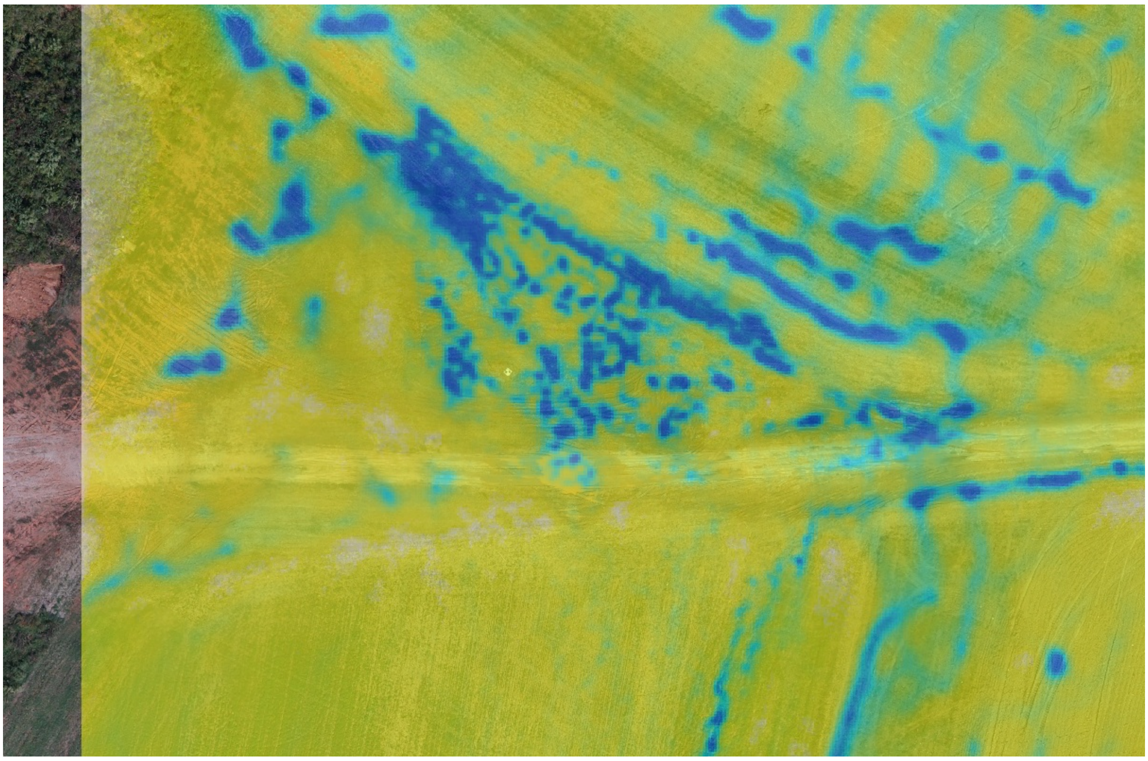
We can validate these results using the orthophotos themselves. The March, June, and both of the October flights were executed after intense rainfalls, so some of the remaining water is still visible. For each of the following two sites, we can first see the puddles in the orthophoto, followed by the depth results overlaying the orthophoto to show where the simulation predicts accumulation will occur.

Orthophoto



Depth Overlaying Orthophoto

Orthophoto



Depth Overlaying Orthophoto