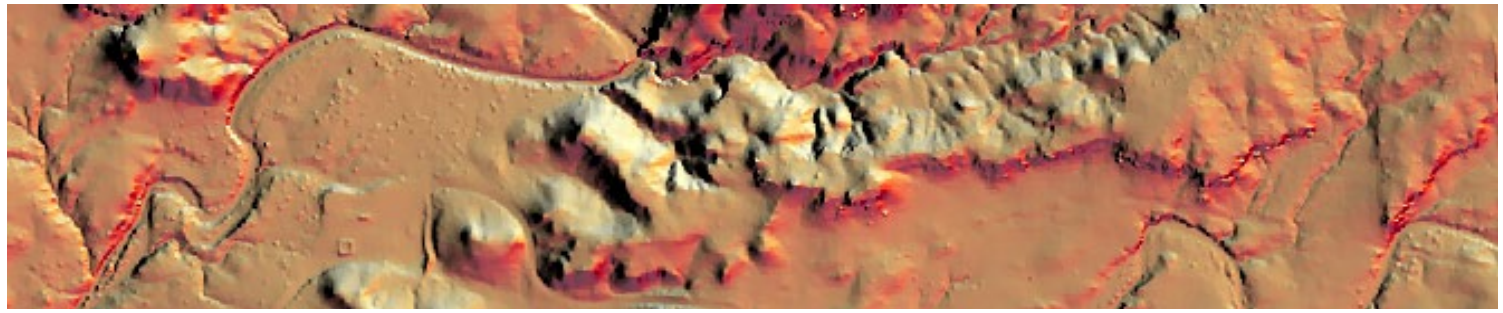


OPTIMIZING SOLAR SITES FOR LCOE THROUGH TERRAIN ANALYSIS

A Geospatial Approach to Solar Tracker Site Design

Matthew Gagne
Lead Renewables Analyst
Geospatial/Energy

KILONEWTON

ACCELERATE YOUR BUSINESS

esri Emerging Business Partner

ABOUT KILONEWTON

Formed in 2017 to solve renewables engineering problems not adequately being addressed

Focus on reducing "soft costs" of renewables, optimizing technology selection & integration, and driving efficiency gains in the design process

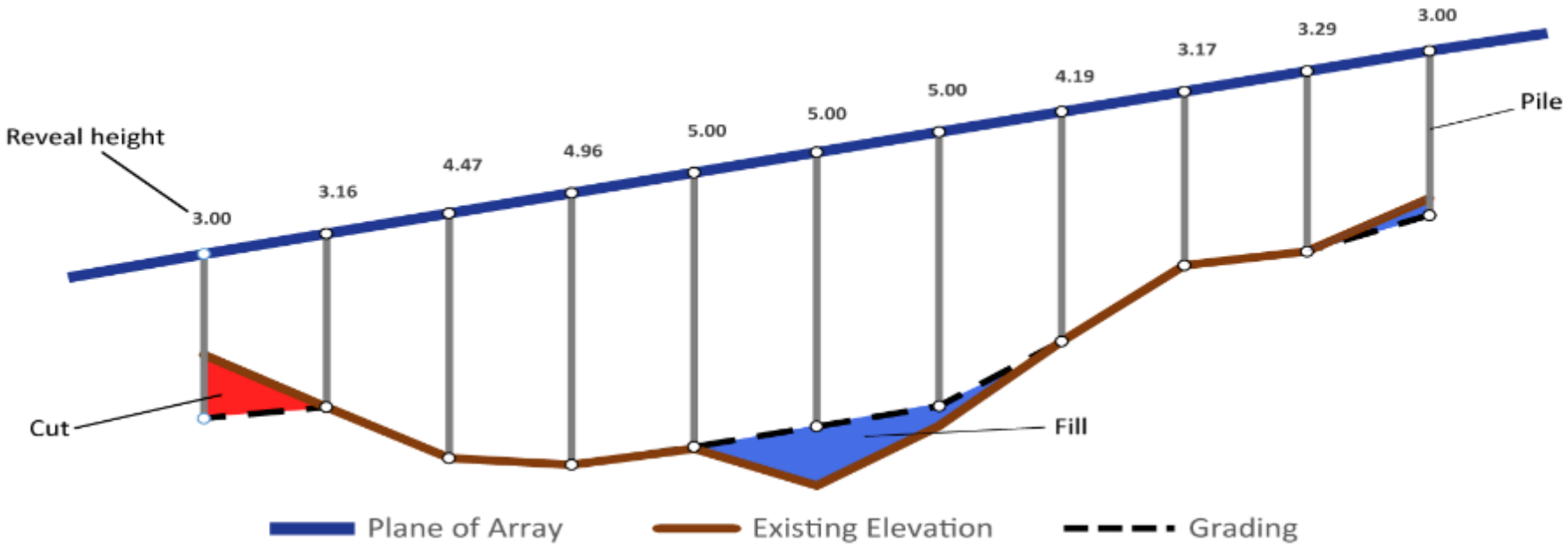
John Williamson
CEO/
Founder

12+ years in solar industry, former Executive Chief Engineer at Array Technologies

Matthew Gagne
Lead Renewables Analyst
Geospatial/Energy

12+ years in solar and wind specializing in geospatial applications and energy analysis

BASIC CONCEPTS



TERMS

Single Axis Tracker

Mechanical Block

Plane of Array

Reveal/Pile

Cut and Fill

MORE COMPLEX TERRAIN

Higher costs for grading and materials

TERRAIN ISSUES ADDRESSED LATE IN DEVELOPMENT

Production losses due to slope of the plane of array

CAD-based design

TRADITIONAL PROCESS NOT CAPABLE OF HANDLING TERRAIN ISSUES EFFECITVELY

Inadequate consideration of terrain

SolarSpace

GIS-based Tools

Developed in ArcGIS Pro & Python

Constantly being developed/improved

Addresses terrain-related design issues before they become a problem

Spatial Analyst

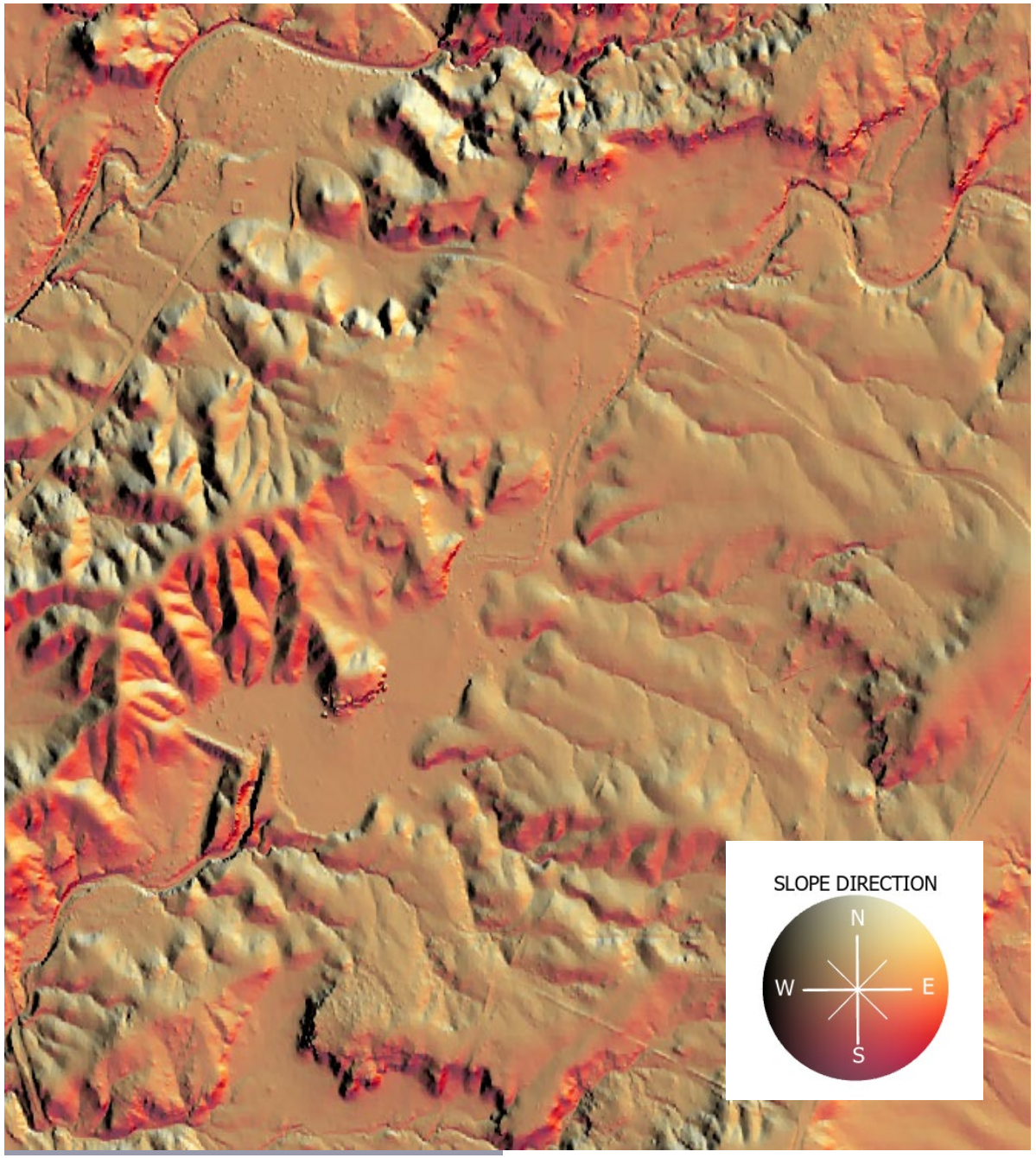
3D Analyst

DIRECTIONAL SLOPE ANALYSIS

Solar tracker losses defined by NSEW terrain and plane of array variations

Cardinal direction slope algorithm

Helps in initial design process



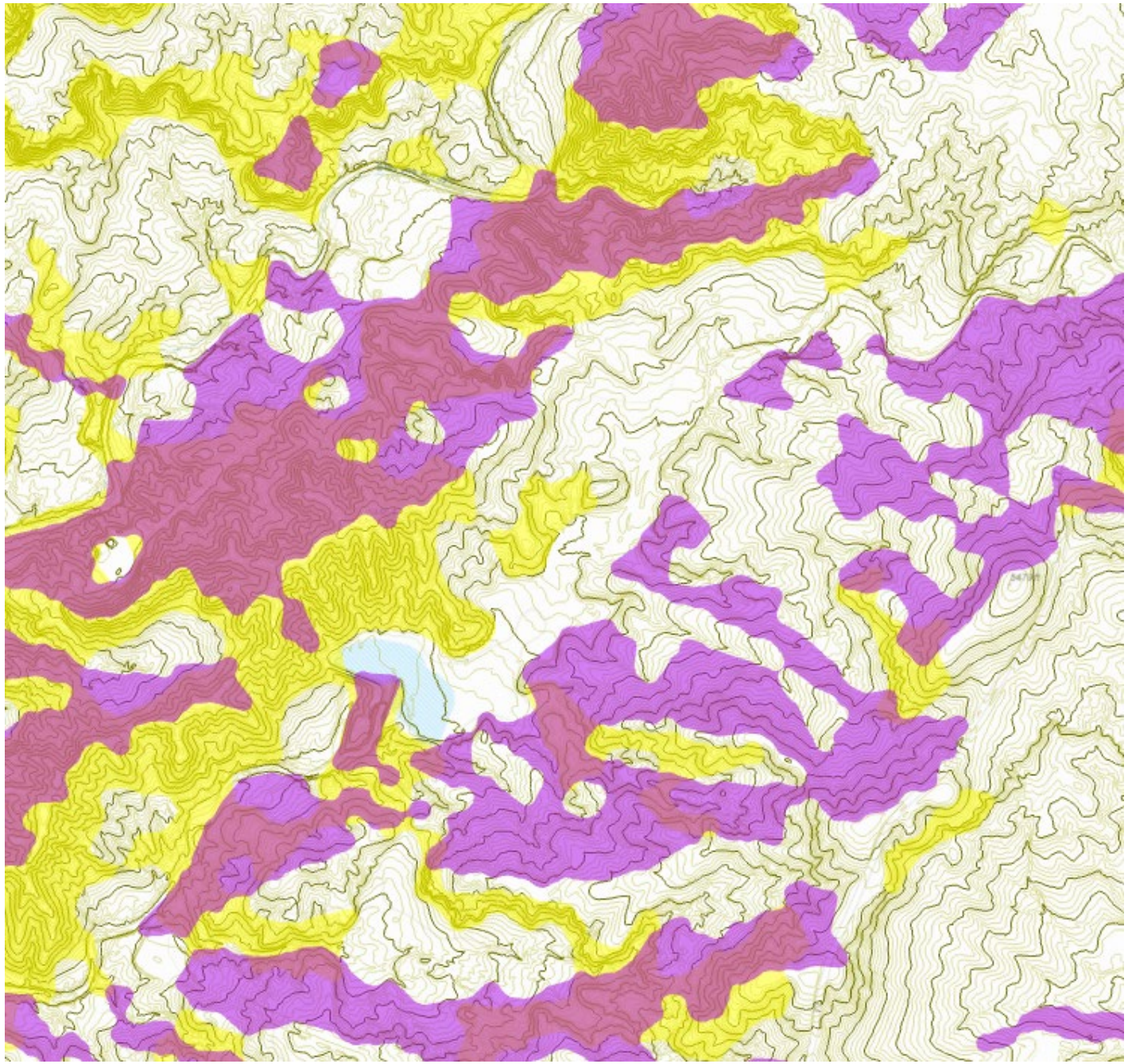
SLOPE EXCLUSION ANALYSIS

Derived from directional slope algorithm

Input for “mechanical” and “production” slope limits

Ignores small variations and focuses on larger trends

Iterates across the site based on tracker design specifications



IDEAL PLANE OF ARRAY

SolarSpace develops an ideal plane of array based on the terrain within and around individual tracker rows

Incorporates blending from row to row to mitigate east-west shading

Ideal plane drives grading

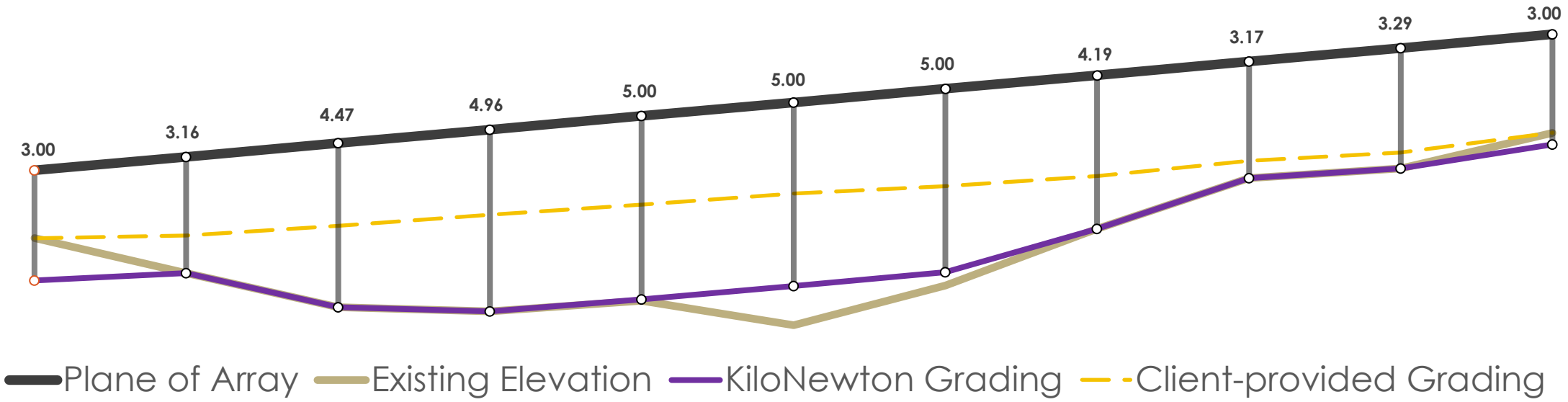
GRADING ANALYSIS

Uses ideal POAs & reveal tolerance to determine grading

Balances cut/fill ratio

Up to 90% less grading than typical plans

Developed to provide full grading plans ready for construction



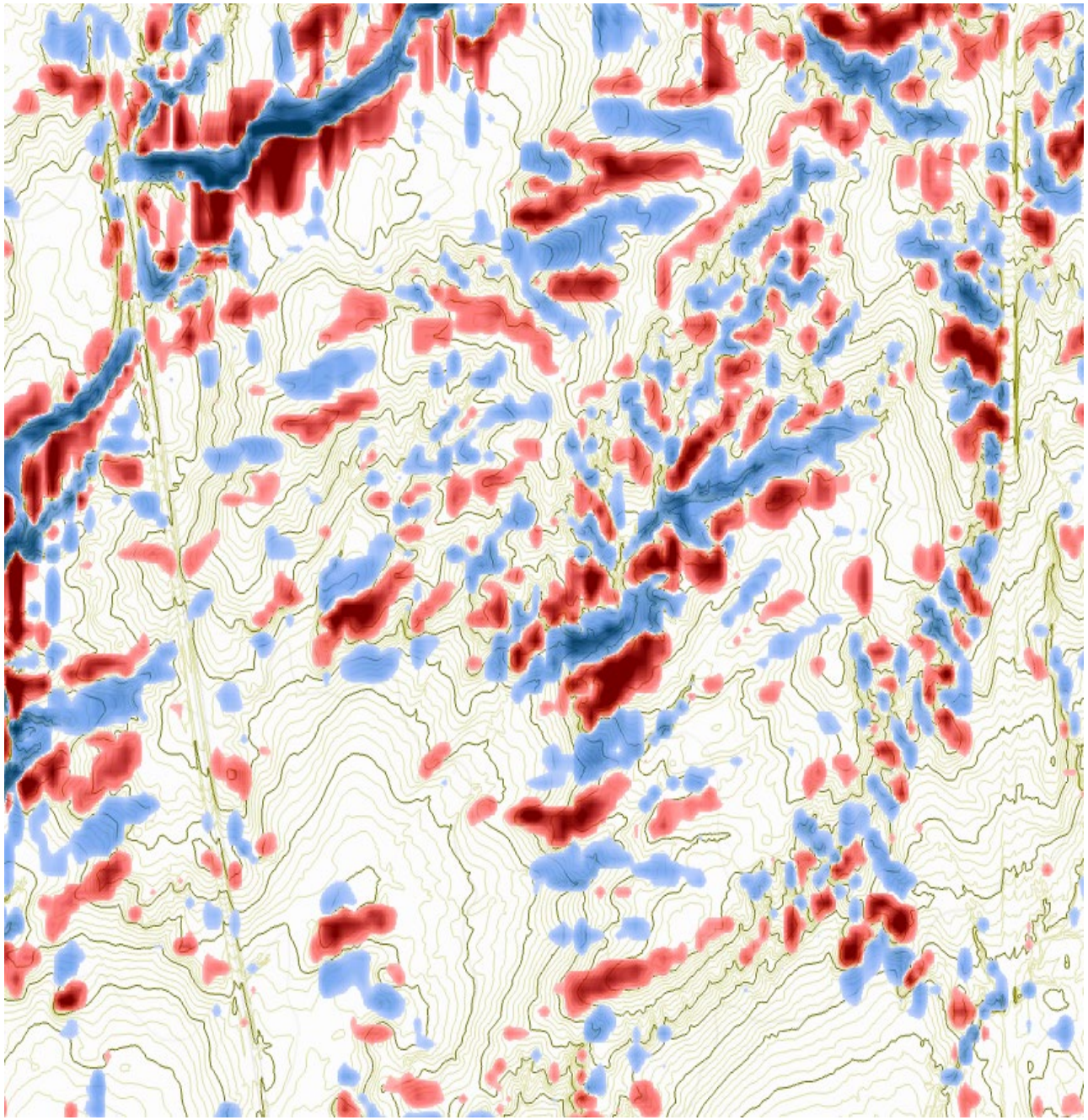
PRELIMINARY GRADING

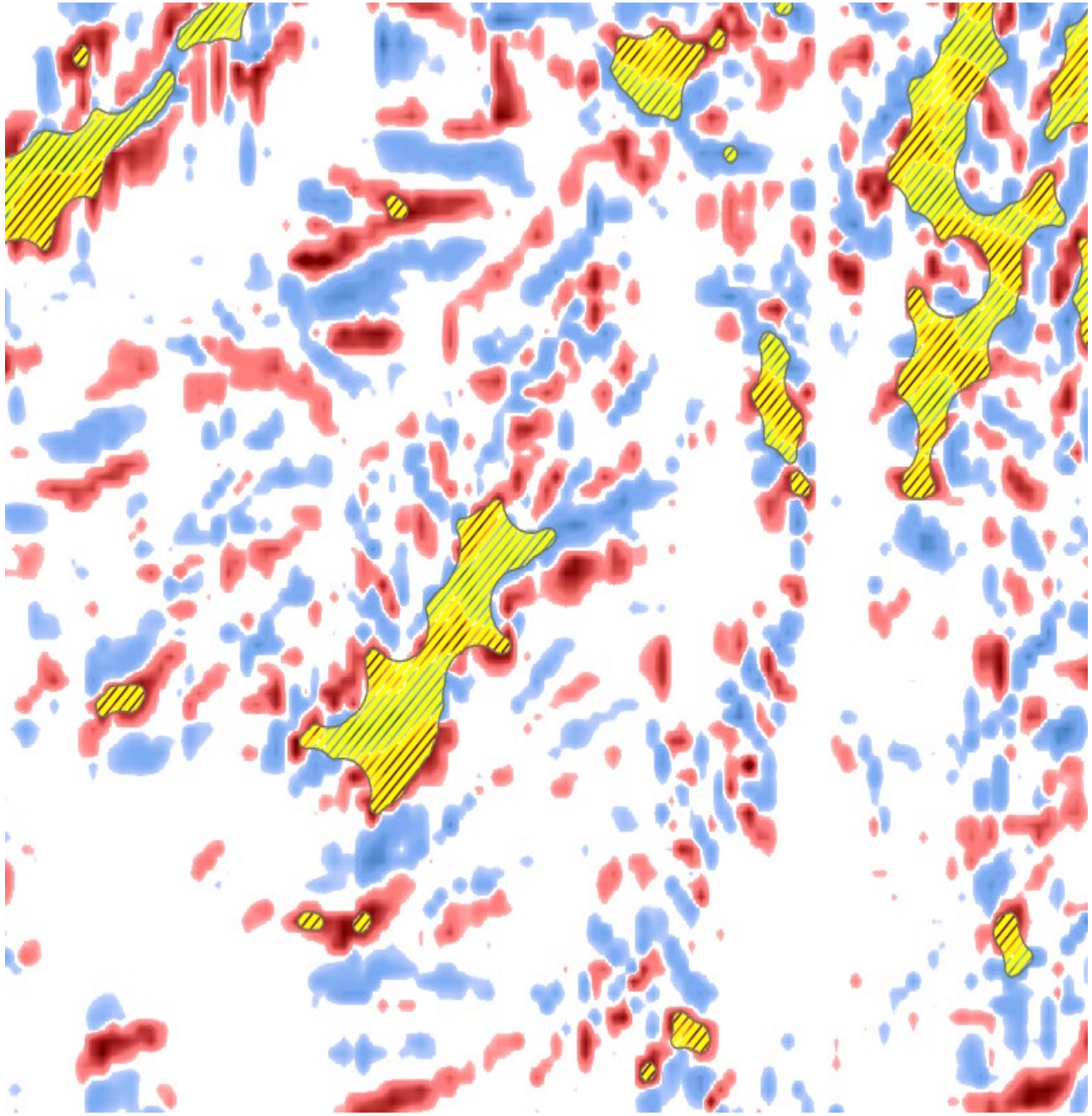
Does not need a layout, just a project boundary

Developed from Site Grading tool

Iterates potential row placement across the site

Provides an estimate that is typically between 2-5% of a fully-designed grading plan





PRELIMINARY GRADING EXCLUSION AREAS

Uses preliminary grading tool to find areas that have the potential for high grading volumes

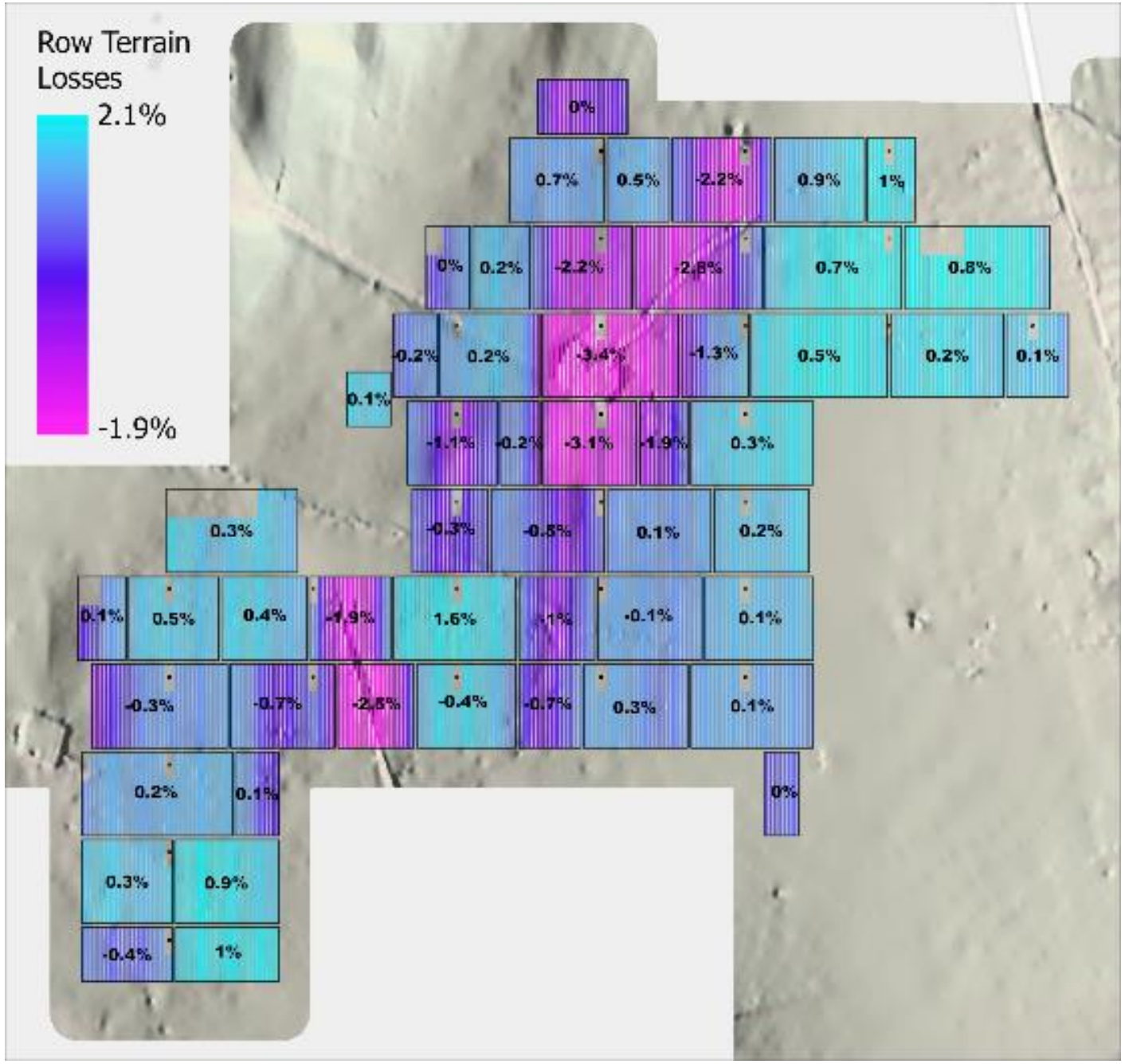
Creates exclusion areas based on pre-defined limits

TERRAIN LOSS ANALYSIS

Uses the Directional Slope and Ideal Plane of Array algorithms

Iterative approach to losses based on angle with respect to latitude

Can be used with designed planes of array or terrain



LCOE ANALYSIS

Uses combinations of all the previous tools and other spatial inputs

Can be done as a preliminary analysis or on a designed layout

Helps identify problem areas and refine designs at any stage of the development process

TOTAL COST / ENERGY PRODUCED

Levelized Cost of Energy (LCOE): The measure of the average net present cost of electricity generation over the lifetime of an energy generating system

CHECKING & OTHER TOOLS

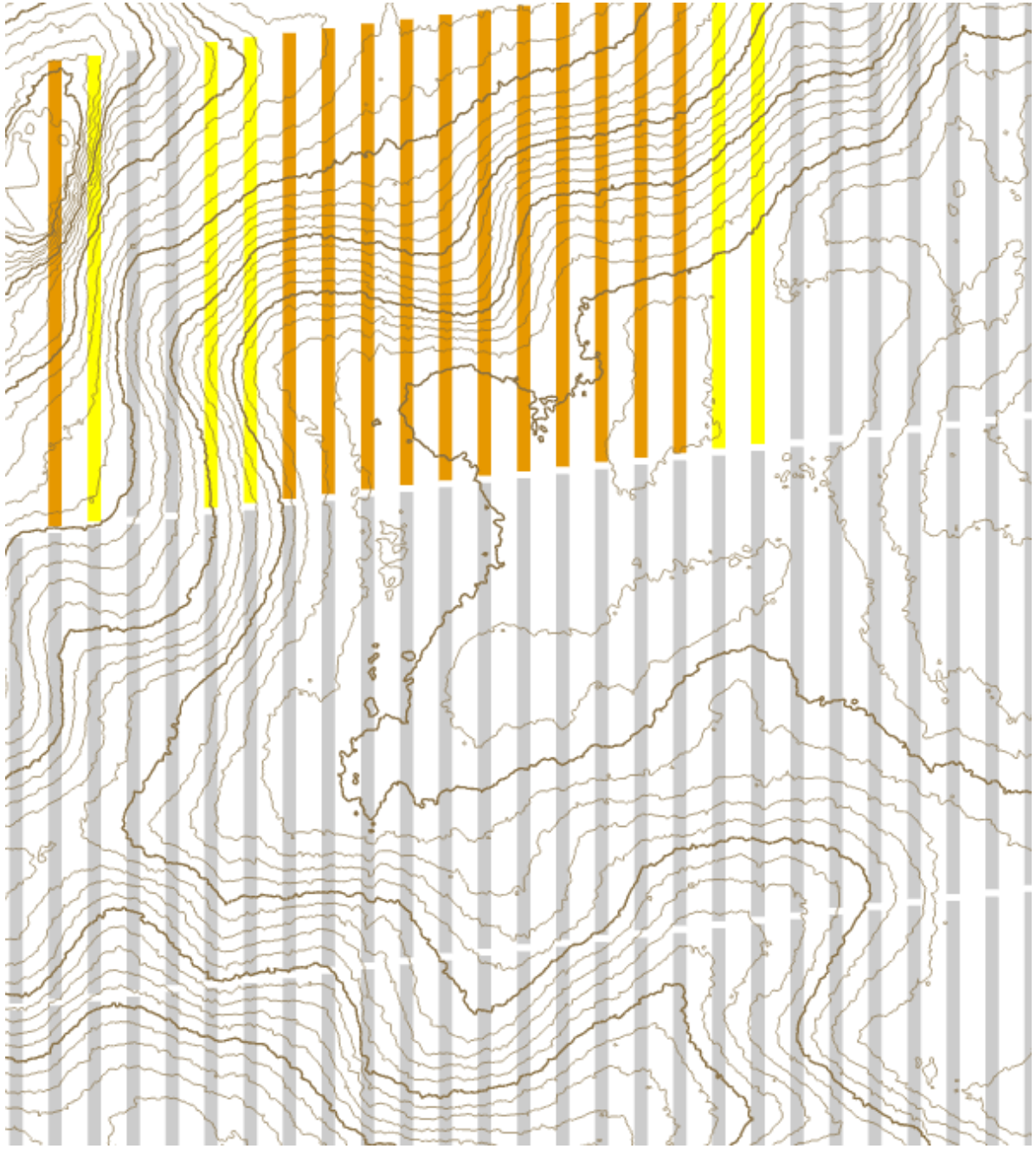
Checking tools have been developed based on site-specific issues

Plane of array north-south slope

Grading slope

Inter-row shading east-west and north-south

Variation in plane of array east-west



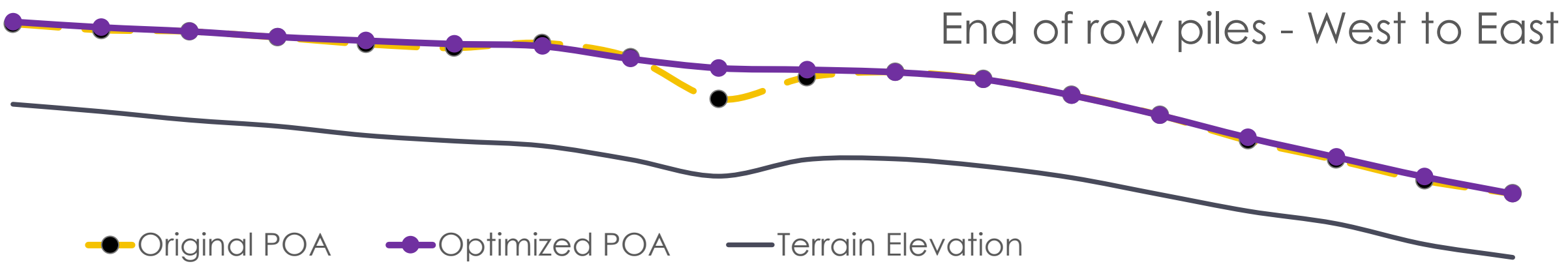
OPTIMIZATION TOOLS

Checking tools evolved into optimization tools to check & solve site-specific issues

Correct for out-of-tolerance plane of array and grading slopes

Plane of array optimization

Eliminate shading east-west/north-south



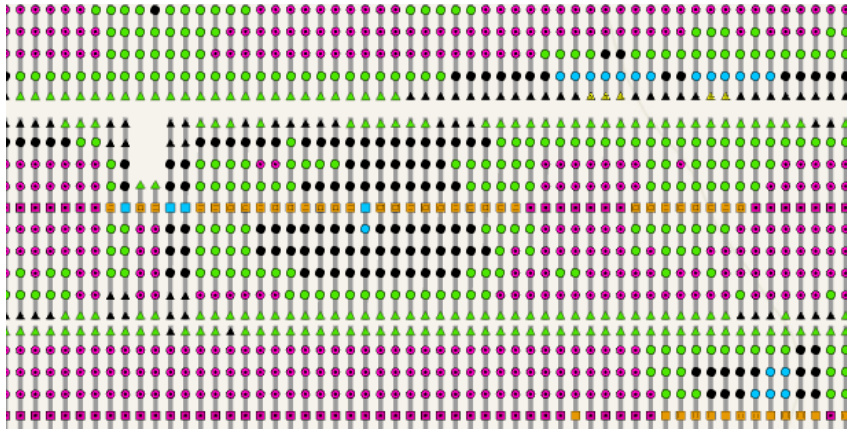
FUTURE DEVELOPMENT

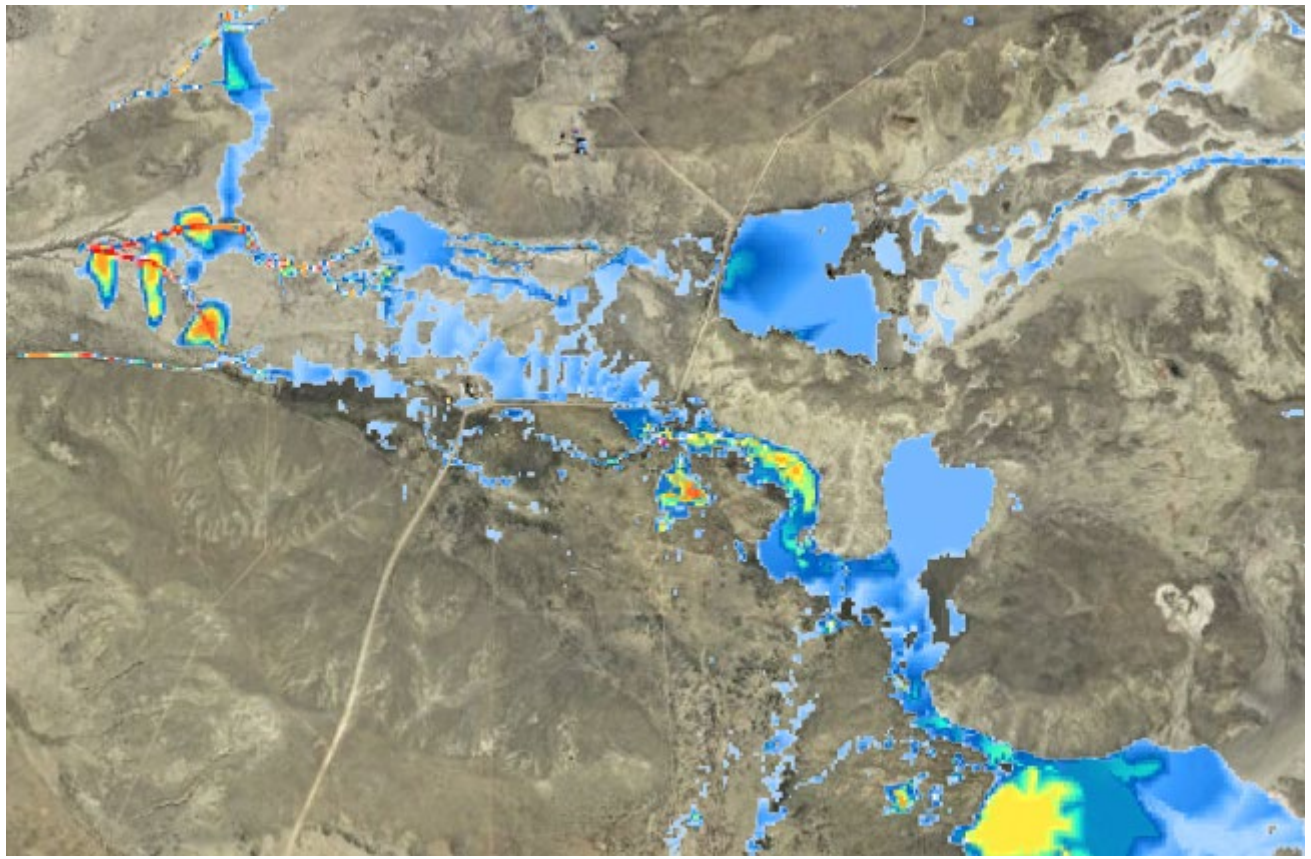
Incorporate hydrology, erosion/deposition, and scour analyses

Incorporate geotechnical data to further enhance and automate pile and foundation design

Online tools for prospecting

```
# Print parameters
arcpy.AddMessage ("Row spacing: " + str(gap_EW) + " ft")
arcpy.AddMessage ("Center to Center Distance: " + str(center

# Define spatial reference
spatialRef = arcpy.Describe(full_area_rows).spatialReference

# Define extents of buildable area
desc = arcpy.Describe(buildable_area)

xMin = str(desc.extent.XMin)
yMin = str(desc.extent.YMin)

originXY = xMin + " " + yMin
yAxisCoor = xMin + " " + str(desc.extent.YMin + 10)

# Create a fishnet of rows
full_area_rows = arcpy.management.CreateFishnet("full_area_r


full_area_rows = rows_final_Sub

# Calculate x and y adjustments to row_area_final polygon ex
x_adj = float ( spacing_EW)/2
y_adj = 1.356610625

# Make this a called script
# sys.path.append(util_path)
```





SUMMARY

Improves site generation on complex sites by up to 5%

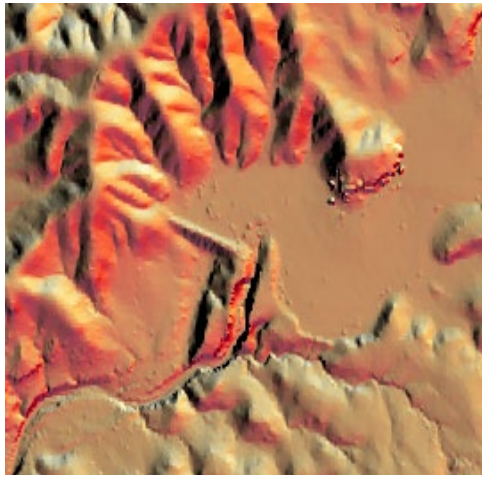
Optimizes layouts to reduce costs by up to 5%

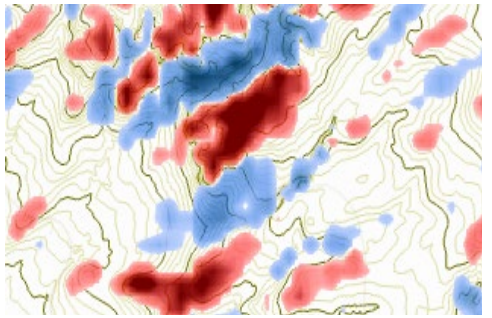
Lowers the cost of electricity by 10%+

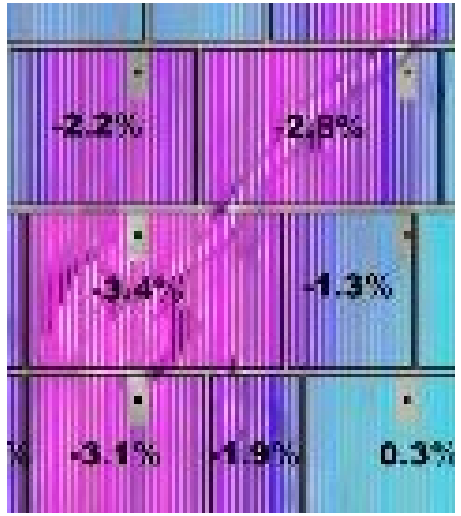
Enables developers to efficiently identify lowest cost sites

Lowers average site costs by up to 20%

Optimizes sites during construction and after by up to 8%







THANK YOU

KILONEWTON

ACCELERATE YOUR BUSINESS

kilonewtonllc.com

matthew@kilonewtonllc.com

esri Emerging Business Partner