From Many User-Contributed Polygons To One Polygon Consensus

Fabian Feitsch

Julius-Maximilians-Universität Würzburg

July 11, 2016

The History of New York

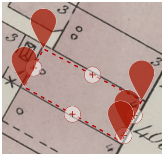
The History of New York

The History of New York

First: Extract the buildings using image processing.

The History of New York

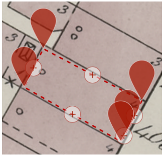
First: Extract the buildings using image processing.



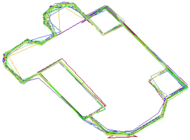
Second: Ask users to fix the extracted polygons.

The History of New York

First: Extract the buildings using image processing.



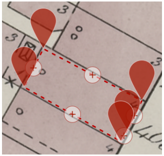
Second: Ask users to fix the extracted polygons.



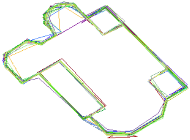
Third: Have a bunch of similar, yet different polygons.

The History of New York

First: Extract the buildings using image processing.



Second: Ask users to fix the extracted polygons.

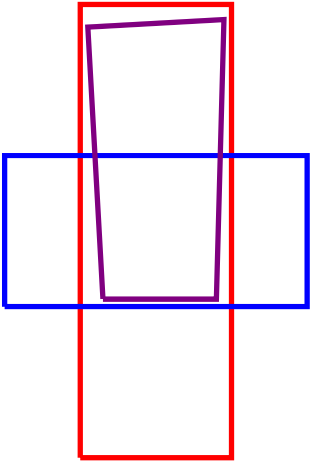


Third: Have a bunch of similar, yet different polygons.

Fourth: Which footprint was meant by the users?

Table Of Contents

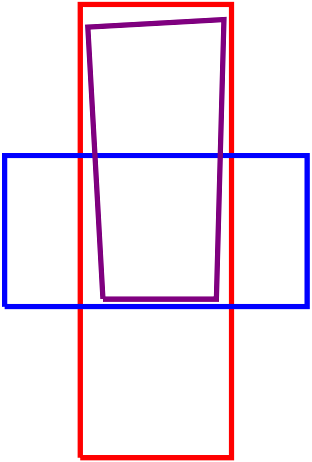
The Pile Algorithm



The Pile Algorithm

Idea

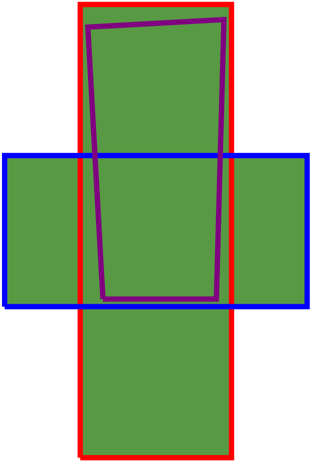
Return the area that is covered by at least k polygons.



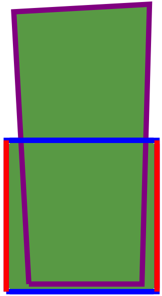
The Pile Algorithm

Idea

Return the area that is covered by at least k polygons.



The Pile Algorithm



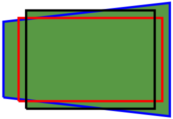
Idea

Return the area that is covered by at least k polygons.

Implications of the Pile Algorithm

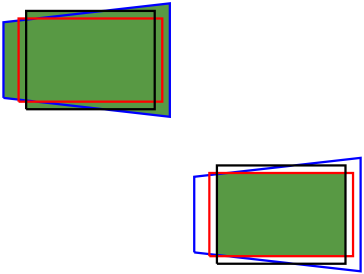
Implications of the Pile Algorithm

- Simple Interpretation



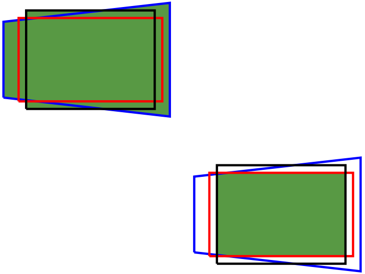
Implications of the Pile Algorithm

- Simple Interpretation
- Takes only one parameter k



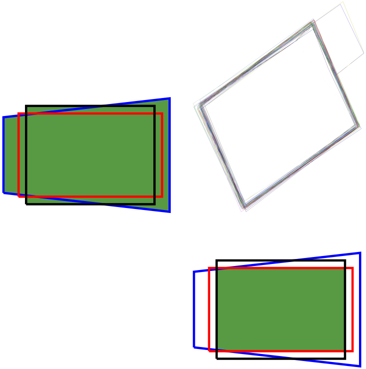
Implications of the Pile Algorithm

- Simple Interpretation
- Takes only one parameter k
- Scaling does not matter



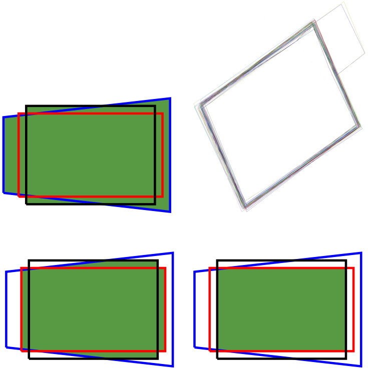
Implications of the Pile Algorithm

- Simple Interpretation
- Takes only one parameter k
- Scaling does not matter
- Implementation difficulties



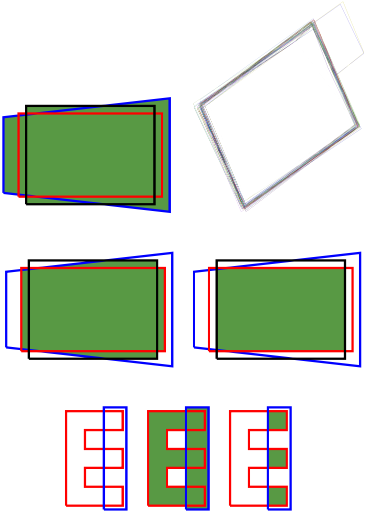
Implications of the Pile Algorithm

- Simple Interpretation
- Takes only one parameter k
- Scaling does not matter
- Implementation difficulties
- Generates too many corners



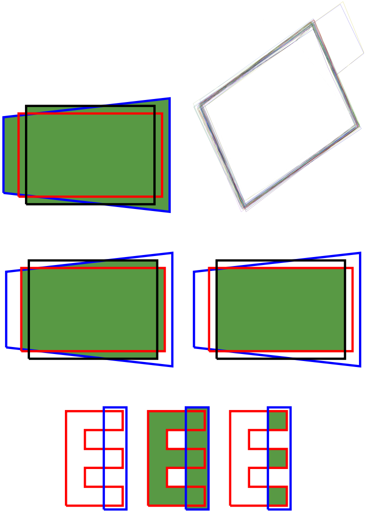
Implications of the Pile Algorithm

- Simple Interpretation
- Takes only one parameter k
- Scaling does not matter
- Implementation difficulties
- Generates too many corners
- Result is not always a simple polygon



Implications of the Pile Algorithm

- Simple Interpretation
- Takes only one parameter k
- Scaling does not matter
- Implementation difficulties
- Generates too many corners
- Result is not always a simple polygon



Conclusion

The Pile algorithm is not useful for the *Building Inspector*.

Strategy of Cluster-Based Algorithms

Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

Strategy of Cluster-Based Algorithms

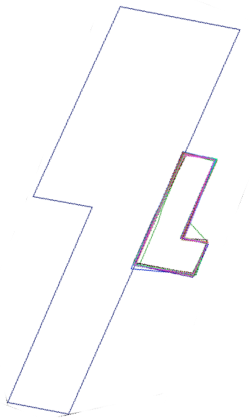
Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters

Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

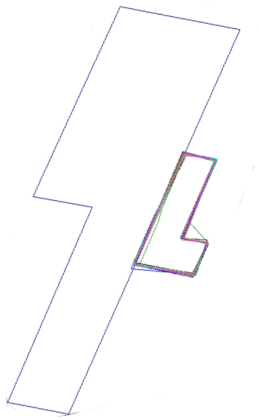
1. Outlier Removal requires parameters



Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters

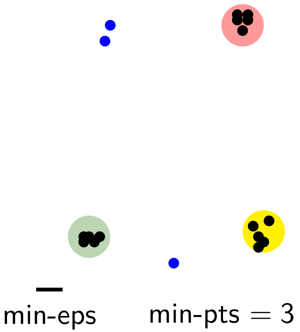


Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters

For example DBSCAN:

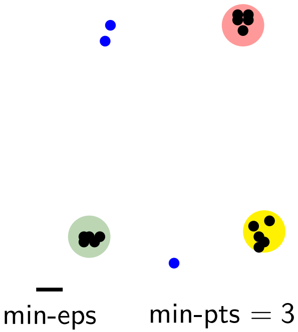


Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm

For example DBSCAN:

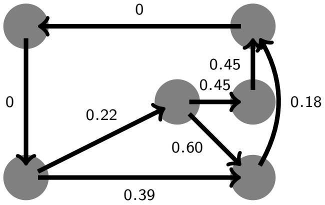


Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm

Generate graph from clusters and extract cycle.

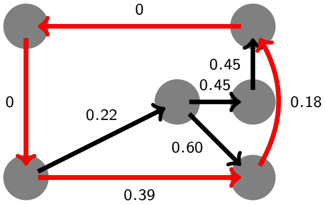


Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm
4. Translate Clusters to Polygon

Generate graph from clusters and extract cycle.



Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm
4. Translate Clusters to Polygon

Use centroids of the clusters in the circle.

Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm
4. Translate Clusters to Polygon

Use centroids of the clusters in the circle.

Claim: Either Step 1 or Step 2 can be omitted.

Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm
4. Translate Clusters to Polygon

Use centroids of the clusters in the circle.

Claim: Either Step 1 or Step 2 can be omitted.
To be shown later . . .

Strategy of Cluster-Based Algorithms

Four steps derived from Mauricio Arteaga (NYPL):

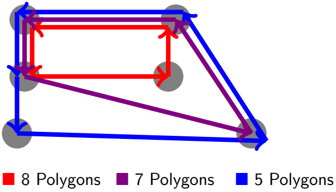
1. Outlier Removal requires parameters
2. Clustering of Corners requires parameters
3. Apply actual Algorithm ← Now!
4. Translate Clusters to Polygon

Use centroids of the clusters in the circle.

Claim: Either Step 1 or Step 2 can be omitted.
To be shown later ...

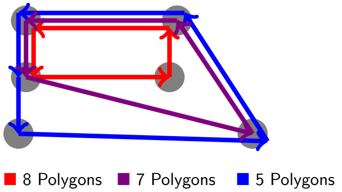
The Voting Algorithm

The Voting Algorithm



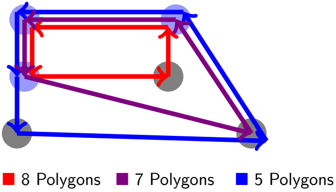
The Voting Algorithm

1. Find cluster c with most corners.



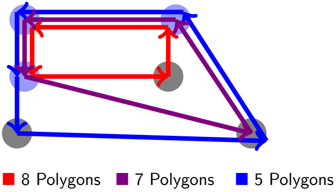
The Voting Algorithm

1. Find cluster c with most corners.



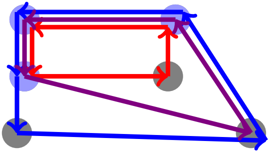
The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.



The Voting Algorithm

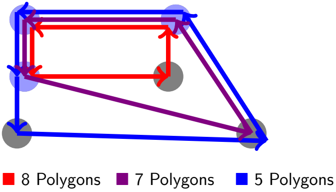
1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.



8 Polygons 7 Polygons 5 Polygons

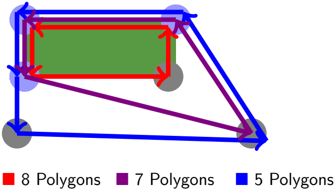
The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.



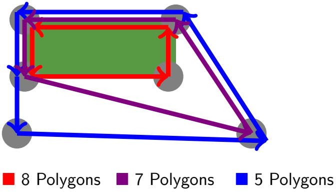
The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.



The Voting Algorithm

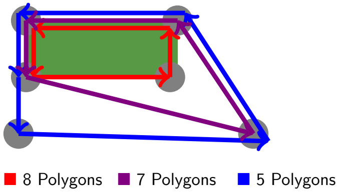
1. Find cluster *c* with most corners.
2. Select neighbor of *c* which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.



Unfortunately, there are subtleties:

The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.

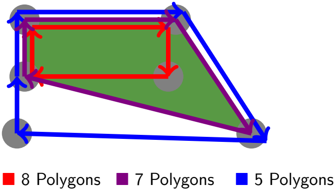


Unfortunately, there are subtleties:

- Orientation of Polygons

The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.

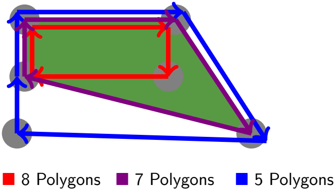


Unfortunately, there are subtleties:

- Orientation of Polygons

The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.

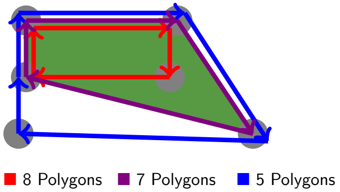


Unfortunately, there are subtleties:

- Orientation of Polygons
- Result can be arbitrarily bad ...

The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.

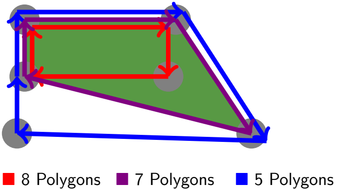


Unfortunately, there are subtleties:

- Orientation of Polygons
- Result can be arbitrarily bad ...
- ... or not existent in the input.

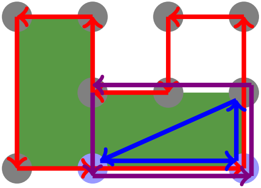
The Voting Algorithm

1. Find cluster c with most corners.
2. Select neighbor of c which most users agree with.
3. Proceed until cycle is found.
4. Return that cycle.



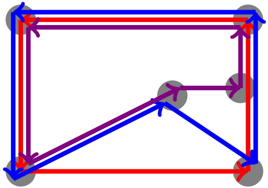
Unfortunately, there are subtleties:

- Orientation of Polygons
- Result can be arbitrarily bad ...
- ... or not existent in the input.



Minimum Mean Weight Cycle Algorithm

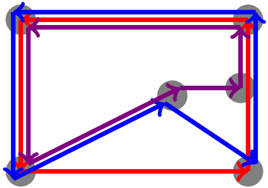
Minimum Mean Weight Cycle Algorithm



■ 8 Polygons ■ 7 Polygons ■ 5 Polygons

Minimum Mean Weight Cycle Algorithm

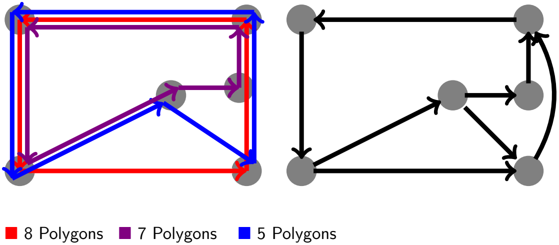
1. Consider input as directed, weighted graph.



8 Polygons 7 Polygons 5 Polygons

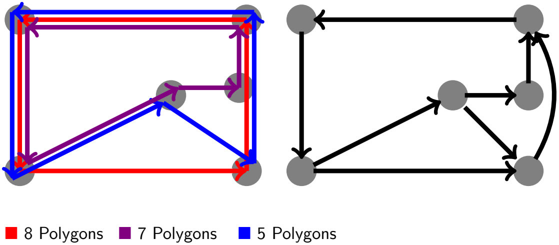
Minimum Mean Weight Cycle Algorithm

1. Consider input as directed, weighted graph.



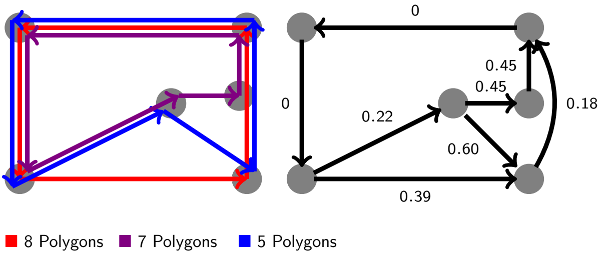
Minimum Mean Weight Cycle Algorithm

1. Consider input as directed, weighted graph.
2. Let k be the count of edges between c_1 and c_2.
   Then the edge (c_1, c_2) has the weight $w = -\log(\frac{k}{n})$.



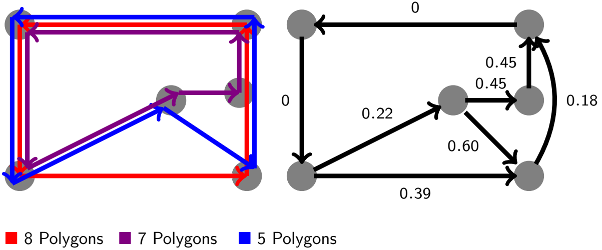
Minimum Mean Weight Cycle Algorithm

1. Consider input as directed, weighted graph.
2. Let k be the count of edges between c_1 and c_2. Then the edge (c_1, c_2) has the weight $w = -\log(\frac{k}{n})$.



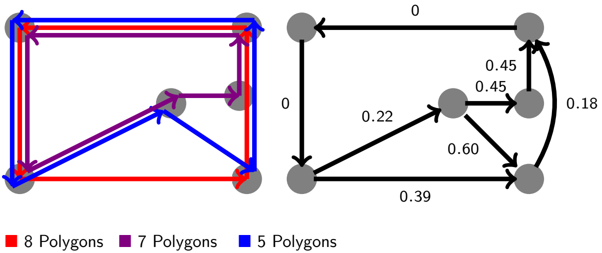
Minimum Mean Weight Cycle Algorithm

1. Consider input as directed, weighted graph.
2. Let k be the count of edges between c_1 and c_2.
   Then the edge (c_1, c_2) has the weight $w = -\log(\frac{k}{n})$.
3. Find cycle with minimum mean weight in the graph.



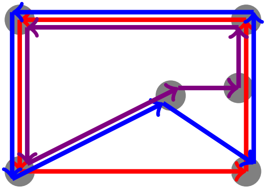
Minimum Mean Weight Cycle Algorithm

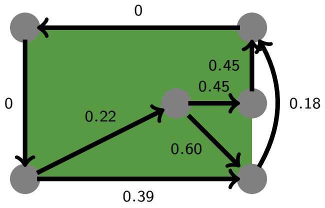
1. Consider input as directed, weighted graph.
2. Let k be the count of edges between c_1 and c_2.
   Then the edge (c_1, c_2) has the weight $w = -\log(\frac{k}{n})$.
3. Find cycle with minimum mean weight in the graph.
4. Return that cycle and translate it to a polygon.



Minimum Mean Weight Cycle Algorithm

1. Consider input as directed, weighted graph.
2. Let k be the count of edges between c_1 and c_2.
   Then the edge (c_1, c_2) has the weight $w = -\log(\frac{k}{n})$.
3. Find cycle with minimum mean weight in the graph.
4. Return that cycle and translate it to a polygon.

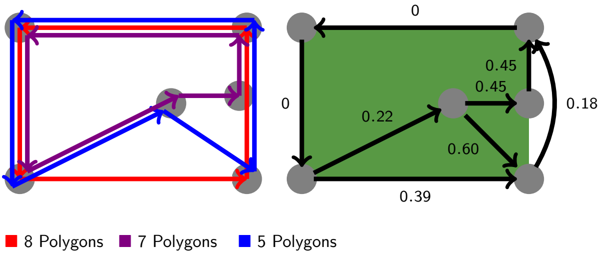




8 Polygons 7 Polygons 5 Polygons

Minimum Mean Weight Cycle Algorithm

1. Consider input as directed, weighted graph.
2. Let k be the count of edges between c_1 and c_2.
   Then the edge (c_1, c_2) has the weight $w = -\log(\frac{k}{n})$.
3. Find cycle with minimum mean weight in the graph.
4. Return that cycle and translate it to a polygon.



No more orientation-dependent!

Choice Of Parameters

Choice Of Parameters

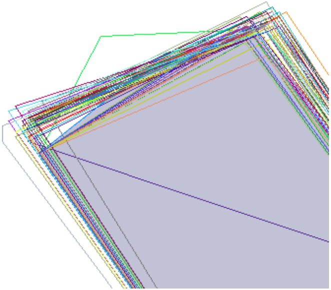
Problem: How to find a good min-eps for clustering?

Choice Of Parameters

Problem: How to find a good min-eps for clustering?
Solution so far: Trial and Error

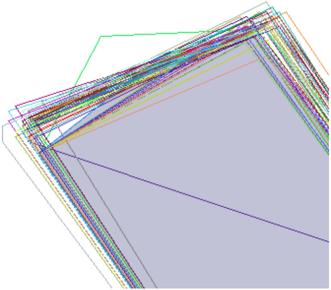
Choice Of Parameters

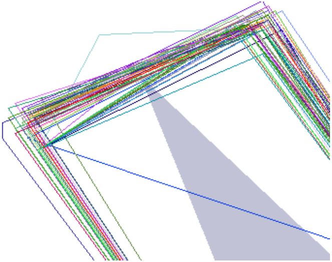
Problem: How to find a good min-eps for clustering?
Solution so far: Trial and Error



Choice Of Parameters

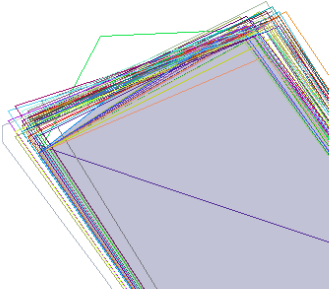
Problem: How to find a good min-eps for clustering?
Solution so far: Trial and Error

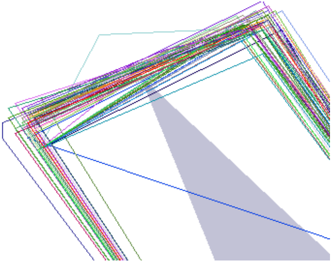




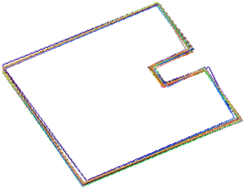
Choice Of Parameters

Problem: How to find a good min-eps for clustering?
Solution so far: Trial and Error
Hope: Find min-eps automatically

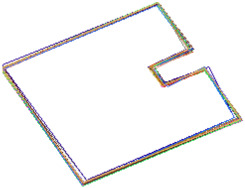


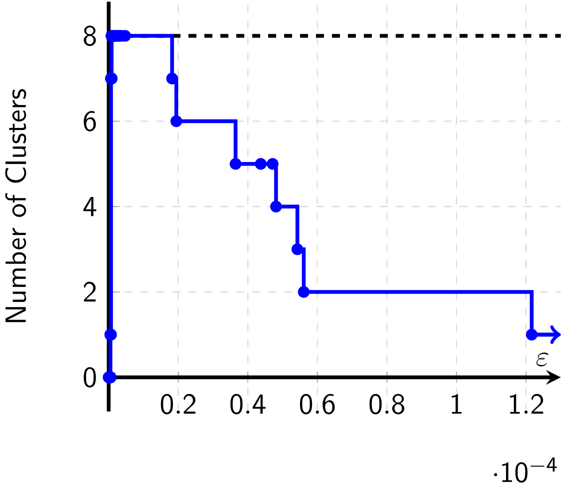


From an Observation to a Heuristic

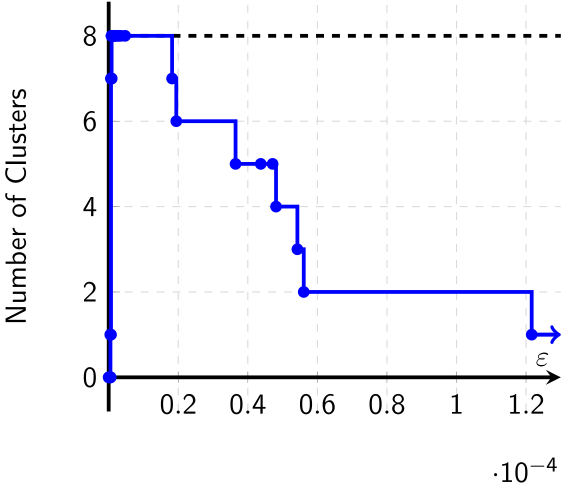


From an Observation to a Heuristic



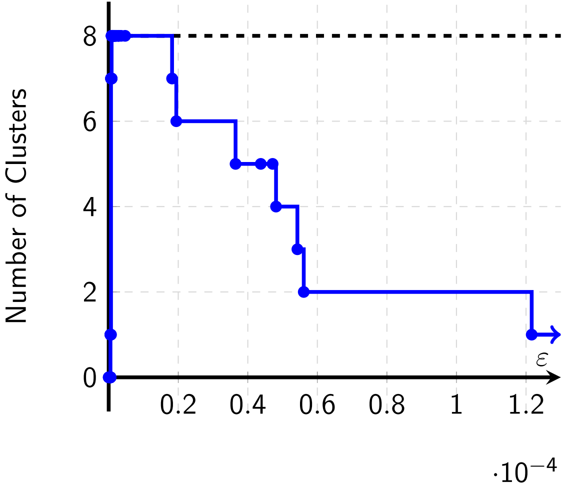


From an Observation to a Heuristic



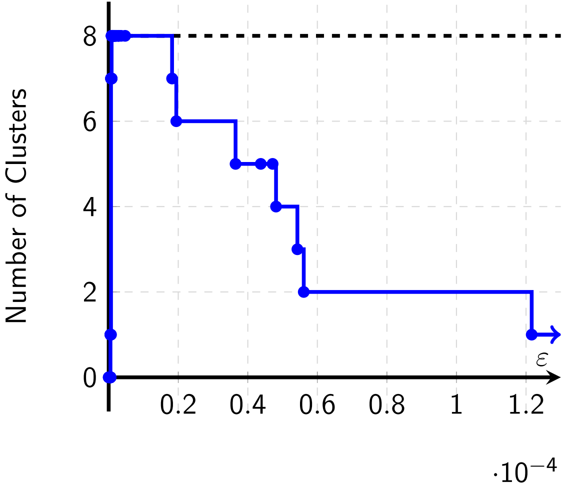
From an Observation to a Heuristic

- Calculate median.



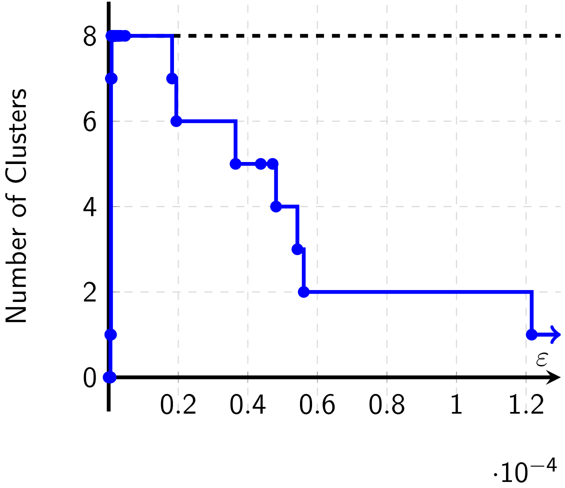
From an Observation to a Heuristic

- Calculate median.
- Allow only clusters of size $n/2$.



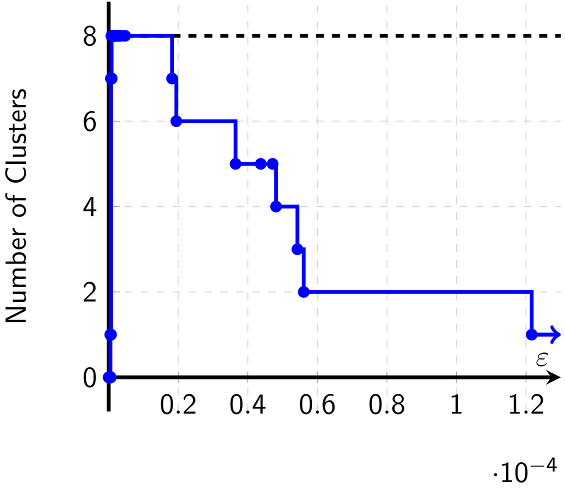
From an Observation to a Heuristic

- Calculate median.
- Allow only clusters of size $n/2$.
- Select longest plateau.



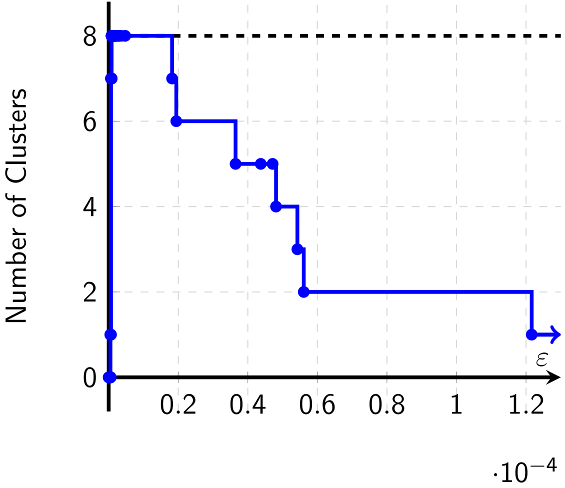
From an Observation to a Heuristic

- Calculate median.
- Allow only clusters of size $n/2$.
- Select longest plateau.
- Use Semi-Dynamic-Sets.



From an Observation to a Heuristic

- Calculate median.
- Allow only clusters of size $n/2$.
- Select longest plateau.
- Use Semi-Dynamic-Sets.
- Running time $\mathcal{O}(n^2 \log n)$.



Evaluation Process

Evaluation Process

Sources

Evaluation Process

Sources

detected,

Evaluation Process

Sources

detected, user,

Evaluation Process

Sources

detected, user,
mmwc-autoeps,

Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw,

Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,

Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps,

Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw,

Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Semantics

Check if the polygons fit to footprints of buildings.

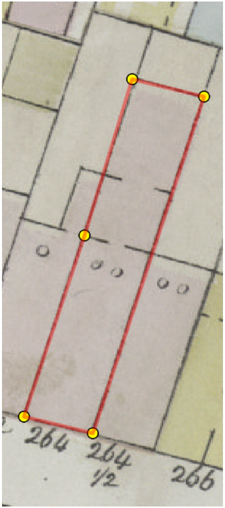
Evaluation Process

Sources

detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Semantics

Check if the polygons fit to footprints of buildings.



Evaluation Process

Sources

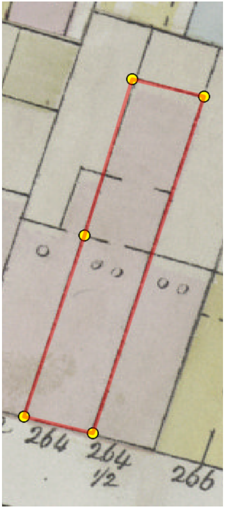
detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Semantics

Check if the polygons fit to footprints of buildings.

Accuracy

Extract the average brightness of the pixels *under* the polygon edges automatically. The lower the better!



Evaluation Process

Sources

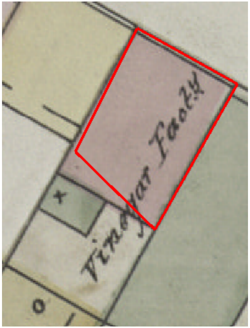
detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Semantics

Check if the polygons fit to footprints of buildings.

Accuracy

Extract the average brightness of the pixels *under* the polygon edges automatically. The lower the better!



Value: 0.48

Evaluation Process

Sources

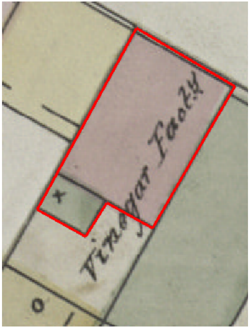
detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Semantics

Check if the polygons fit to footprints of buildings.

Accuracy

Extract the average brightness of the pixels *under* the polygon edges automatically. The lower the better!



Value: 0.38

Evaluation Process

Sources

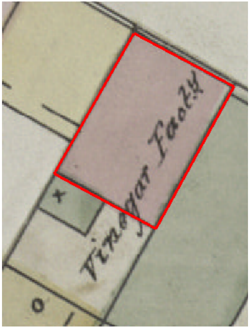
detected, user,
mmwc-autoeps, mmwc-raw, mmwc-clean,
voting-autoeps, voting-raw, voting-clean

Semantics

Check if the polygons fit to footprints of buildings.

Accuracy

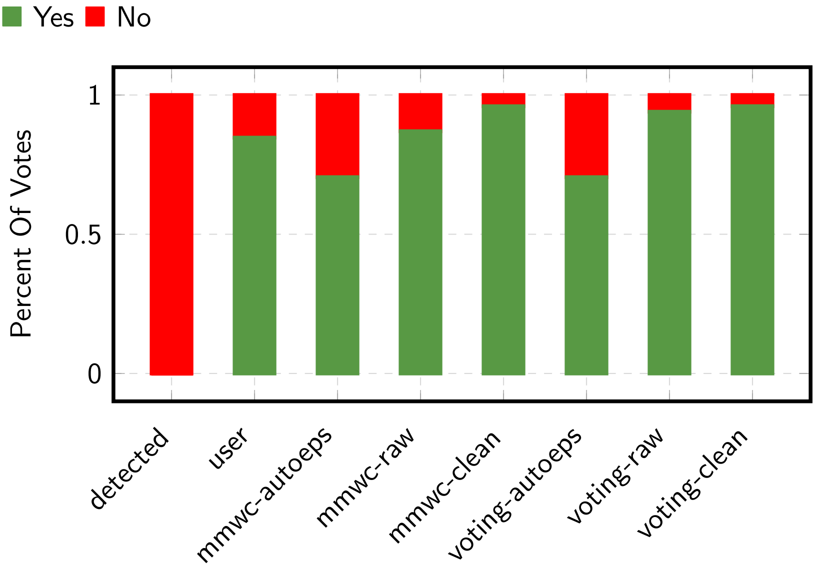
Extract the average brightness of the pixels *under* the polygon edges automatically. The lower the better!



Value: 0.36

Sematical Results

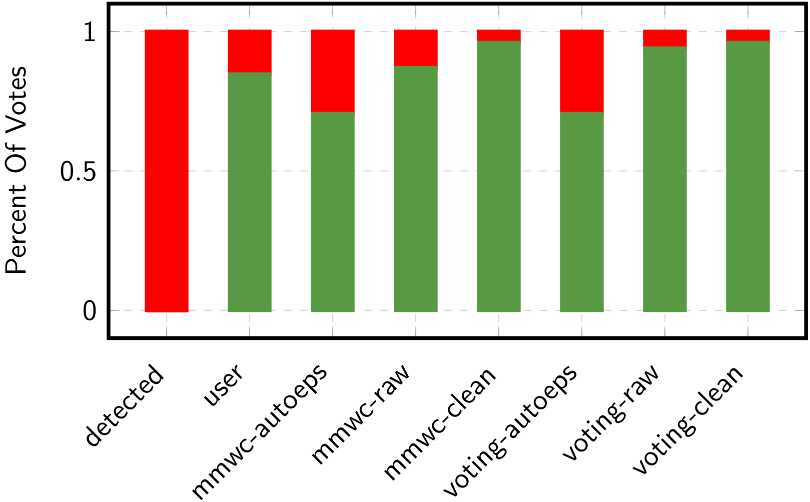
Sematical Results



Sematical Results

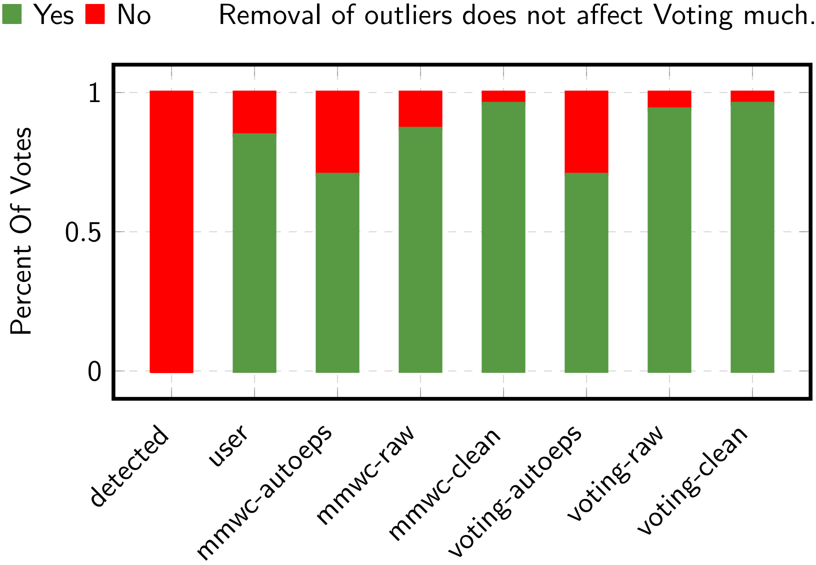
Yes No

MMWC was not better than Voting!



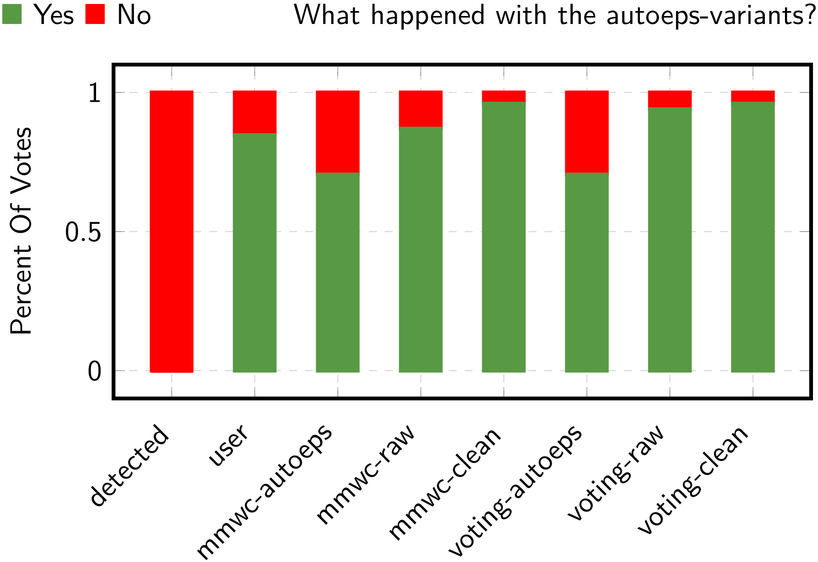
Sematical Results

Removal of outliers does not affect Voting much.



Sematical Results

What happened with the autoeps-variants?



Analysis of the auto-eps Variants

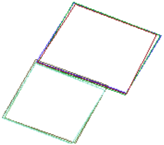
Analysis of the auto-eps Variants

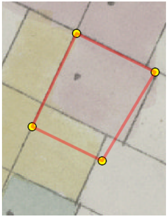
There were 45 groups where both autoeps-variants failed.

Analysis of the auto-eps Variants

There were 45 groups where both autoeps-variants failed.

Example:

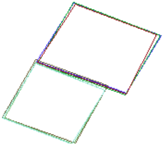


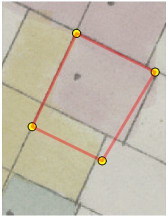


Analysis of the auto-eps Variants

There were 45 groups where both autoeps-variants failed.

Example:



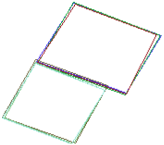


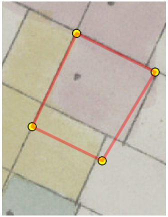
They got confused!

Analysis of the auto-eps Variants

There were 45 groups where both autoeps-variants failed.

Example:





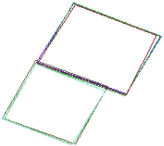
They got confused!

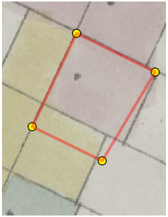
There were two groups only the autoeps-variants could solve.

Analysis of the auto-eps Variants

There were 45 groups where both autoeps-variants failed.

Example:

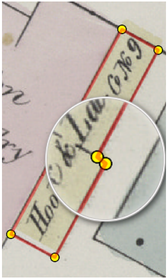


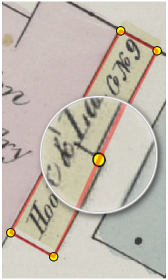


They got confused!

There were two groups only the autoeps-variants could solve.

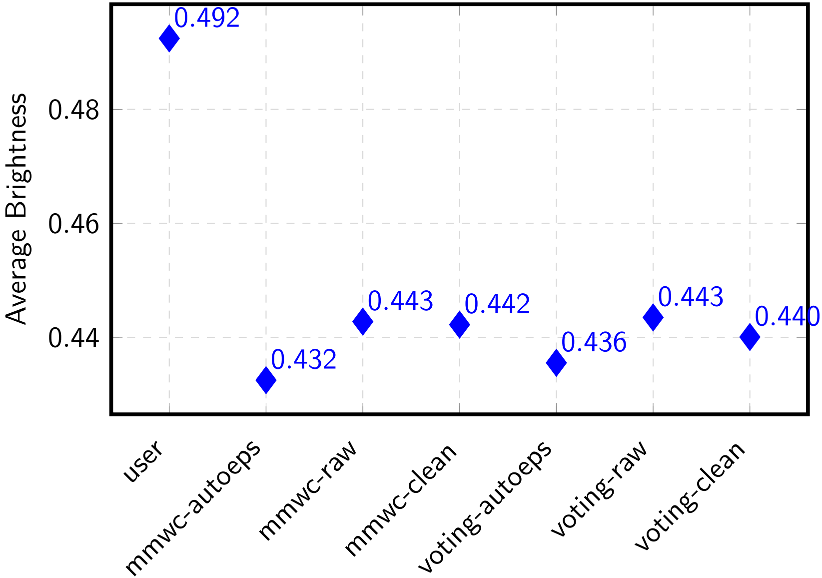
Example:





Brightness Results

Brightness Results



Conclusion

Conclusion

Algorithmically

Conclusion

Algorithmically

We proposed three Algorithms to solve the problem, using pure geometry or cluster-based approaches.

Conclusion

Algorithmically

We proposed three Algorithms to solve the problem, using pure geometry or cluster-based approaches.

Implications from the Results

Conclusion

Algorithmically

We proposed three Algorithms to solve the problem, using pure geometry or cluster-based approaches.

Implications from the Results

The cluster-based algorithms need two pre-processing steps.

Conclusion

Algorithmically

We proposed three Algorithms to solve the problem, using pure geometry or cluster-based approaches.

Implications from the Results

The cluster-based algorithms need two pre-processing steps. One of them can be spared without loss of quality.

Conclusion

Algorithmically

We proposed three Algorithms to solve the problem, using pure geometry or cluster-based approaches.

Implications from the Results

The cluster-based algorithms need two pre-processing steps. One of them can be spared without loss of quality. This choice can be made with regard to the data.

Conclusion

Algorithmically

We proposed three Algorithms to solve the problem, using pure geometry or cluster-based approaches.

Implications from the Results

The cluster-based algorithms need two pre-processing steps. One of them can be spared without loss of quality. This choice can be made with regard to the data.

The Voting Algorithm performs better than the MWWC Algorithm.