The Area Method in the Wolfram Language

13th International Conference on Automated Deduction in Geometry

Jack Heimrath, Wolfram Alpha LLC

What is the Area Method?

The area method is a decision procedure for a subset of Euclidean geometry, developed by Chou, Gao, and Zhang. Its main feature, cited by its authors as an advantage over other decision procedures for geometry, is that it generates shorter, more human-readable proofs of many theorems when compared to other methods . This is achieved by carefully keeping track of the construction steps used when setting up a particular geometric construction and by only allowing conjectures of the form $E_1 = E_2$ (or analogous inequalities), where E_1 and E_2 are arithmetic expressions in geometric quantities . Points occurring in the conjecture are then “eliminated” in the reverse order of their construction using appropriate Elimination Lemmas .

Geometric Quantities

Primitive Objects

- The AM makes use of one type of primitive object, called *points*. These are typically identified with points on the Euclidean plane. The set of all points is denoted $\mathbb{P}$. For our purposes we may identify $\mathbb{P}$ with the Euclidean plane.
- There is also a primitive binary function $\overline{\cdot\cdot} : \mathbb{P}^2 \to \mathbb{R}$, called the *signed distance* , which has the following properties:
  • $\overline{xy} = 0 \Longleftrightarrow x = y$;
  • $\overline{xy} = -\overline{yx}$.
- The final primitive object is a ternary function $\mathcal{S}_{\dots} : \mathbb{P}^3 \to \mathbb{R}$, called the *signed area,* with the following properties:
  • $\mathcal{S}_{abc} = \mathcal{S}_{bca}$;
  • $\mathcal{S}_{abc} = -\mathcal{S}_{acb}$;
  • $\mathcal{S}_{aac} = 0$.

Derived Functions

- For any three points a, b, $c \in \mathbb{P}$ we define their *Pythagorean difference* $\mathcal{P}_{xyz} = \overline{xy}^2 + \overline{yz}^2 - \overline{xz}^2$. This defini - tion has many useful properties, in particular:
  • $\mathcal{P}_{xyz} = 0 \Longleftrightarrow \angle xyz = \frac{\pi}{2}$;

• $\mathcal{P}_{xyx} = \mathcal{P}_{yxy} = 2 \times \overline{xy}^2$;
• $\mathcal{P}_{xxy} = \mathcal{P}_{xyy} = 0$.

- For notational convenience the following two functions are also introduced:
  • $\mathcal{P}_{vxyz} := \mathcal{P}_{vxz} - \mathcal{P}_{yxz}$;
  • $\mathcal{S}_{vxyz} := \mathcal{S}_{vxy} + \mathcal{S}_{vzy}$.

Geometric Quantities

By a *geometric quantity* we mean either of the following:

• a ratio of signed distances $\frac{\overline{ab}}{\overline{xy}}$, subject to the constraints that the points *x* and *y* are distinct and the lines *ab* and *xy* are parallel;
• a signed area or Pythagorean difference.

Elementary Construction Steps

There are 5 Elementary Construction Steps (ECS’s) utilised by the area method. For a construction step to be well-defined certain conditions, called non-degeneracy conditions, may be required. Some ECS’s also require a parameter to be provided as an argument- this may be a real number or a symbolic parameter *r*. Combined, these steps can be used to reproduce a large subset of classical straightedge and compass constructions. Additionally, some non-classical constructions are also possible thanks to the fact that the parameter *r* can be any real number. This makes it possible to, for instance, square the circle.

Elementary Construction Step 1

Constructs an arbitrary point U, denoted ECS1[U]. This is the only ECS which can be invoked without first defining any other points, and as such is used to initialize any and all geometric constructions. Points introduced by this step are called “free points”. All free points in a construction are assumed to be distinct.

In[78]:=
```
ECS1[u]
```
Out[78]= ECS1[u]

It is possible to define many free points at once:

In[79]:=
```
ECS1[x,y,z]
```
Out[79]= ECS1[x, y, z]

Elementary Construction Step 2

Constructs a point Y such that it is the intersection of (LINE U V) and (LINE P Q), denoted by ECS2[y, u, v, p, q].

In[80]:=
```
ECS2[y,u,v,p,q]@ECS1[u,v,p,q]
```
Out[80]=
```
ECS2[y, u, v, p, q][ECS1[u, v, p, q]]
```

Elementary Construction Step 3

Constructs a point Y such that it is the foot from a given point P to (LINE U V), denoted ECS3[y, p, u, v],

In[81]:=
```
ECS3[y,p,u,v]@ECS1[p,u,v]
```
Out[81]=
```
ECS3[y, p, u, v][ECS1[p, u, v]]
```

Elementary Construction Step 4

Constructs a point Y on the line passing through the point W and parallel to (LINE U V) such that $\overline{WY} = r\,\overline{UV}$, denoted ECS4[y, w, u, v, r]. Note that r can be a real number, a geometric quantity, or a variable.

In[82]:=
```
ECS4[y,w,u,v,r]@ECS1[w,u,v]
```
Out[82]=
```
ECS4[y, w, u, v, r][ECS1[w, u, v]]
```

Elementary Construction Step 5

Constructs a point Y on the line passing through the point U and perpendicular to (LINE U V) such that $\frac{4\,S_{UVY}}{\mathcal{P}_{UVU}} = r$, denoted ECS5[y, u, v, r]. Note that r can be a real number, a geometric quantity, or a variable.

In[83]:=
```
ECS5[y,u,v,r]@ECS1[u,v]
```
Out[83]=
```
ECS5[y, u, v, r][ECS1[u, v]]
```

Geometric Constructions

A geometric construction is a (finite) list $C = (C_1, C_2, \ldots, C_n)$ where each C_i, $1 \le i \le n$ is an Elementary Construction Step. For each C_i the points used in that construction step must already by introduced by some construction step appearing earlier in the list. The point introduced by step i is said to have order i within the construction.

Elimination Lemmas

Elimination Lemmas are used to eliminate all occurrences of constructed points from the stated conjecture. At each step this elimination process removes the last- with respect to the construction order- constructed point which still occurs in the conjecture. In many cases this procedure will be sufficient to resolve a conjecture. For example the triangle inequality can be constructed and stated using only free points, which means that no Elimination Lemmas can be applied. However, we still expect the triangle inequality to be true and provable. In cases like this one it is necessary to make use of area coordinates.

In cases like this it is necessary to make use of a area coordinates. In total there are 13 Elimination Lemmas and an additional 3 which can be applied if the use of area coordinates is necessary.

Why use Wolfram Language?

- There are many reasons why the Wolfram Language is well suited for the area method. A particularly important one is that Wolfram Language excels at (conditional) pattern matching. This makes it very quick and easy to implement and apply the point elimination procedure described above. The example below shows that the internal representation of an elimination lemma is essentially a direct translation of its mathematical form.

In[84]:=
```
Column[{
    UnderBar["Elimination  Lemma  9"],
    "If  Y  is  introduced  by  ECS5[Y,P,Q,r],  then:",
    TraditionalForm [Subscript[𝒫,  ABY]==Subscript[𝒫,  ABP]-4r  Subscript[𝒮,  PAQB]],
    ,
    UnderBar["Internal  Implementation "],
    Block[
        {Y,P,Q,r},
        (*{p=construction [Y,"points "]⟦1⟧,construction [Y,"points "]⟦2⟧,r=construction [Y,"paramete
        HoldPattern [PythagoreanDifference  [A_,B_,Y,p0_Association ]]:→PythagoreanDifference  [A,B
    ]
    }]
```

Out[84]=

Elimination Lemma 9

If Y is introduced by ECS5[Y,P,Q,r], then:

$\mathcal{P}_{ABY} = \mathcal{P}_{ABP} - 4\, r\, \mathcal{S}_{PAQB}$

Internal Implementation

```
HoldPattern[PythagoreanDifference [A_, B_, Y, p0_Association]] :→
 PythagoreanDifference [A, B, P, p0] - 4 r SignedArea[P, A, Q, B, p0]
```

- Another reason is accessibility - all one needs to prove theorems is internet access.
- This might also be useful in the classroom.

Examples

The Existence of the Euler Line

The Euler Line is a line determined for any triangle which is not equilateral. It passes through many triangle centers, in particular through the orthocenter, circumcenter, and centroid.

In[85]:=
```
Get["C:\\Users\\heimr\\WolframWorkspaces\\Kernel\\StartUp\\PlaneGeometry\\AreaMethod.m"];
```

Get : Cannot open C:\Users\heimr\WolframWorkspaces\Kernel\StartUp\PlaneGeometry\AreaMethod.m.

In[86]:=
```
(* Constructing the orthocenter, circumcenter, and centroid of a triangle. *)
ClearAll[ELConstruction];
ELConstruction=IsOrthocenter[x,a,b,c]@IsCircumcenter[y,a,b,c]@IsCentroid[z,a,b,c]@FreePoint
```

Out[87]=
```
IsOrthocenter[x, a, b, c][
 IsCircumcenter[y, a, b, c][IsCentroid[z, a, b, c][FreePoint[a, b, c]]]]
```

In[88]:=
```
(* Stating the conjecture. *)
ClearAll[ELConjecture];
ELConjecture=(Collinear[x,y,z]//TraditionalForm)
```

Out[89]//TraditionalForm=

Collinear(x, y, z)

In[90]:=
```
(* Proving the conjecture. *)
Column[AbsoluteTiming@VerifyConjecture[ELConjecture,ELConstruction,"UseAreaCoordinates"→Tr
```

$7. \times 10^{-6}$

Out[90]=
```
VerifyConjecture[Collinear(x, y, z), IsOrthocenter[x, a, b, c][
  IsCircumcenter[y, a, b, c][IsCentroid[z, a, b, c][FreePoint[a, b, c]]]],
 UseAreaCoordinates → True, ProofNotebook → True]
```

Note that the way we stated the conjecture above it is also true for equilateral triangles, since in that case $\mathcal{S}_{xyz}$ is also zero.

The Triangle Inequality

In[91]:=
```
(* Constructing a triangle. *)
ClearAll[triangle];
triangle=FreePoint[a,b,c]
```

Out[92]=
```
FreePoint[a, b, c]
```

In[93]:=
```
(* Stating the conjecture. *)
ClearAll[triangleInequality];
triangleInequality=(Sqrt[PythagoreanDifference[a,b,a]/2]+Sqrt[PythagoreanDifference[b,c,b]/2]≥
```

Out[94]//TraditionalForm=

$$\frac{\sqrt{\text{PythagoreanDifference}\ (a,\ b,\ a)}}{\sqrt{2}}+\frac{\sqrt{\text{PythagoreanDifference}\ (b,\ c,\ b)}}{\sqrt{2}}\geq\frac{\sqrt{\text{PythagoreanDifference}\ (a,\ c,\ a)}}{\sqrt{2}}$$

In[95]:=
```
(* Proving the inequality . *)
VerifyConjecture [triangleInequality ,triangle ,"UseAreaCoordinates "→True ,"ProofNotebook "→Tru
```

Out[95]=

$$\text{VerifyConjecture}\left[\frac{\sqrt{\text{PythagoreanDifference}(a, b, a)}}{\sqrt{2}} + \frac{\sqrt{\text{PythagoreanDifference}(b, c, b)}}{\sqrt{2}} \geq \frac{\sqrt{\text{PythagoreanDifference}(a, c, a)}}{\sqrt{2}}, \text{FreePoint}[a, b, c], \text{UseAreaCoordinates} \to \text{True}, \text{ProofNotebook} \to \text{True}\right]$$

Ceva’s Theorem

In[96]:=
```
(* Ceva 's Theorem Construction *)
ClearAll [cevasSetup ];
cevasSetup =IsIntersection [f,c,o,a,b]@IsIntersection [e,b,o,a,c]@IsIntersection [d,a,o,b,c]@Fr
```

Out[97]=
```
IsIntersection [f, c, o, a, b][
 IsIntersection [e, b, o, a, c][IsIntersection [d, a, o, b, c][FreePoint [a, b, c, o]]]]
```

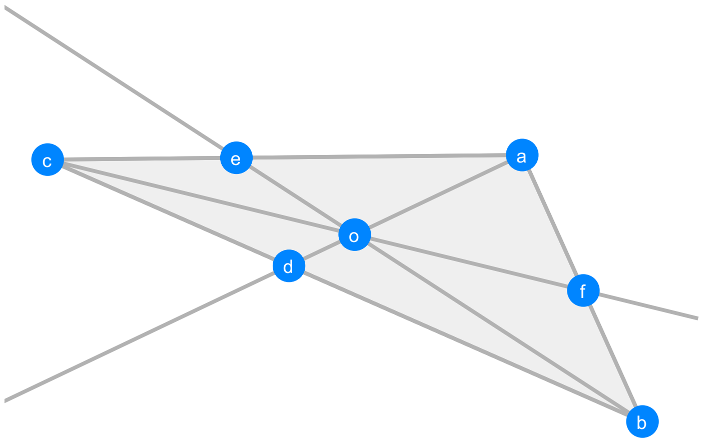
In[98]:=
```
(* Stating the conjecture . *)
ClearAll [cevasTheorem ];
```

$$\text{cevasTheorem} = \left(\frac{\text{SignedDistance}[a,f]}{\text{SignedDistance}[f,b]}\frac{\text{SignedDistance}[b,d]}{\text{SignedDistance}[d,c]}\frac{\text{SignedDistance}[c,e]}{\text{SignedDistance}[e,a]} == 1\right) \text{//TraditionalForm}$$

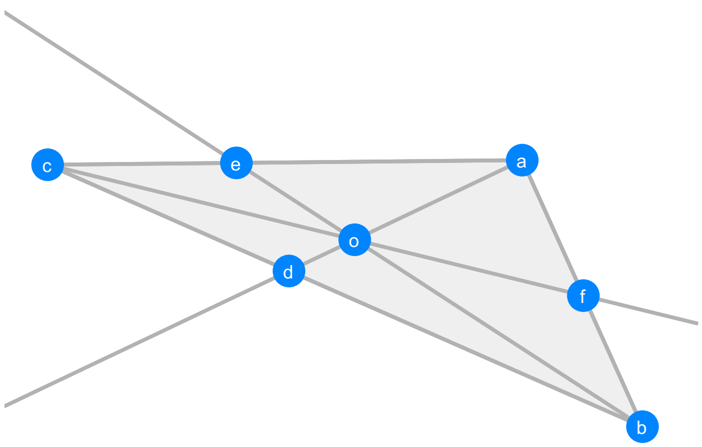
Out[99]//TraditionalForm=

$$\frac{\text{SignedDistance}(a, f)\,\text{SignedDistance}(b, d)\,\text{SignedDistance}(c, e)}{\text{SignedDistance}(e, a)\,\text{SignedDistance}(f, b)\,\text{SignedDistance}(d, c)} = 1$$

In[100]:=



Out[100]=



In[101]:=

```
VerifyConjecture [cevasTheorem ,cevasSetup ,"ProofNotebook "→True]
```

Out[101]=

$$\text{VerifyConjecture}\left[\frac{\text{SignedDistance}(a, f)\,\text{SignedDistance}(b, d)\,\text{SignedDistance}(c, e)}{\text{SignedDistance}(e, a)\,\text{SignedDistance}(f, b)\,\text{SignedDistance}(d, c)} = 1,\right.$$

IsIntersection [f, c, o, a, b][IsIntersection [e, b, o, a, c][

IsIntersection [d, a, o, b, c][FreePoint [a, b, c, o]]]], ProofNotebook → True]

In[102]:=
```
(* What if we try to prove a false conjecture ? *)
ClearAll [cevaCounterexample ];
cevaCounterexample =(cevasTheorem /.SignedDistance [a,f]→SignedDistance [a,c])
```

Out[103]//TraditionalForm=

$$\frac{\text{SignedDistance}(a,\ c)\ \text{SignedDistance}(b,\ d)\ \text{SignedDistance}(c,\ e)}{\text{SignedDistance}(e,\ a)\ \text{SignedDistance}(f,\ b)\ \text{SignedDistance}(d,\ c)} = 1$$

In[104]:=
```
Grid[{
    AbsoluteTiming @VerifyConjecture [cevaCounterexample ,cevasSetup ,"ProofNotebook "→True],
    AbsoluteTiming @VerifyConjecture [cevaCounterexample ,cevasSetup ,"ProofNotebook "→True ,"C
}]
```

Out[104]=
```
6. × 10^-6   VerifyConjecture [SignedDistance (a,c) SignedDistance (b,d) SignedDistance (c,e) / SignedDistance (e,a) SignedDistance (f,b) SignedDistance (d,c) = 1,
               IsIntersection [f, c, o, a, b][IsIntersection [e, b, o, a, c][
                 IsIntersection [d, a, o, b, c][FreePoint[a, b, c, o]]]], ProofNotebook → True]
1. × 10^-6   VerifyConjecture [SignedDistance (a,c) SignedDistance (b,d) SignedDistance (c,e) / SignedDistance (e,a) SignedDistance (f,b) SignedDistance (d,c) = 1,
               IsIntersection [f, c, o, a, b][IsIntersection [e, b, o, a, c][
                 IsIntersection [d, a, o, b, c][FreePoint[a, b, c, o]]]],
               ProofNotebook → True, CheckParallelRatios → True]
```

Some Evaluation Times

In[105]:=
```
(* Set up some constructions and conjectures . *)
ClearAll[cevasTheorem , desarguesTheorem , ELTheorem ,
  gaussNewtonLineTheorem , heronsFormula , interceptTheorem , midpointTheorem ,
  menelausTheorem , pappusTheorem , pythagorasTheorem , triangleInequality ];
cevasSetup = IsIntersection [...][...] ;
cevasTheorem =
  TraditionalForm [((AreaMethod`SignedDistance [a, f] × AreaMethod`SignedDistance [b, d]) ×
        AreaMethod`SignedDistance [c, e]) /
      ((AreaMethod`SignedDistance [f, b] × AreaMethod`SignedDistance [d, c]) ×
        AreaMethod`SignedDistance [e, a]) == 1];
desarguesSetup = OnInterLineParallel [...][...] ;
desarguesTheorem = Parallel[b, c, b2, c2] // TraditionalForm ;
ELTheoreomSetup = IsCentroid [...][...] ;
ELTheorem = TraditionalForm [AreaMethod`Collinear [x, y, z]];
gaussNewtonLineSetup = IsMidpoint [...][...] ;
gaussNewtonLineTheorem = TraditionalForm [AreaMethod`Collinear [m1, m2, m3]];
heroSetup = FreePoint [...] ;
heronsFormula = TraditionalForm [16 AreaMethod`SignedArea [a, b, c]^2 ==
```

```
    AreaMethod`PythagoreanDifference [a, c, a] × AreaMethod`PythagoreanDifference [
        b, a, b] - AreaMethod`PythagoreanDifference [c, a, b]^2];
interceptSetup = IsIntersection [...][...] + ;
interceptTheorem = TraditionalForm[
    AreaMethod`SignedDistance [s, a] / AreaMethod`SignedDistance [a, b] == AreaMethod`SignedDistance [s, c] / AreaMethod`SignedDistance [c, d]];
midpointSetup = IsMidpoint [...][...] + ;
midpointTheorem = TraditionalForm [AreaMethod`Parallel [a, b, d, e]];
menelausSetup = IsIntersection [...][...] + ;
menelausTheorem =
  TraditionalForm [((AreaMethod`SignedDistance [a, f] × AreaMethod`SignedDistance [b, d]) ×
        AreaMethod`SignedDistance [c, e]) /
      ((AreaMethod`SignedDistance [f, b] × AreaMethod`SignedDistance [d, c]) ×
        AreaMethod`SignedDistance [e, a]) == -1];
pappusSetup = IsIntersection [...][...] + ;
pappusTheorem = TraditionalForm [AreaMethod`Collinear [x, y, z]];
pythagoreanSetup = OnPerp [...][...] + ;
pythagorasTheorem = (*TraditionalForm [AreaMethod`PythagoreanDifference [c,a,b]==0]*)
  AreaMethod`PythagoreanDifference [c, a, b] == 0;
triangleInequalitySetup = ECS1 [...] + ;
triangleInequality = Sqrt[AreaMethod`PythagoreanDifference [a, b, a] / 2] +
    Sqrt[AreaMethod`PythagoreanDifference [b, c, b] / 2] ≥
   Sqrt[AreaMethod`PythagoreanDifference [c, a, c]] / 2;
Grid[{{"Theorem", "Area Coordinates", "Time", "Result"},
  Flatten @{"Ceva's Theorem", "no", AbsoluteTiming [
     VerifyConjecture [cevasTheorem , cevasSetup (*,"CheckParallelRatios "→True*)]]},
  Flatten @{"Desargues Theorem", "no", AbsoluteTiming [
     VerifyConjecture [desarguesTheorem , desarguesSetup ]]},
  Flatten @{"Euler Line", "yes", AbsoluteTiming [
     VerifyConjecture [ELTheorem , ELTheoreomSetup , "UseAreaCoordinates " → True]]},
  Flatten @{"Gauss-Newton Line", "yes", AbsoluteTiming [VerifyConjecture [
      gaussNewtonLineTheorem , gaussNewtonLineSetup , "UseAreaCoordinates " → True]]},
  Flatten @{"Heron's Formula", "yes", AbsoluteTiming [
     VerifyConjecture [heronsFormula , heroSetup , "UseAreaCoordinates " → True]]},
  Flatten @{"Intercept Theorem", "no", AbsoluteTiming [VerifyConjecture [
      interceptTheorem , interceptSetup , "CheckParallelRatios " → True]]},
  Flatten @{"Midpoint Theorem", "no", AbsoluteTiming [VerifyConjecture [
      midpointTheorem , midpointSetup ]]}, Flatten @{"Menelaus ' Theorem",
    "no", AbsoluteTiming [VerifyConjecture [menelausTheorem , menelausSetup ]]},
  Flatten @{"Pappus's Line Theorem", "no",
```

```
    AbsoluteTiming [VerifyConjecture [pappusTheorem , pappusSetup ]]},
  Flatten @{"Pythagorean  Theorem ", "no",
    AbsoluteTiming [VerifyConjecture [pythagorasTheorem , pythagoreanSetup ]]},
  Flatten @{"Triangle  Inequality ", "yes", AbsoluteTiming [VerifyConjecture [
      triangleInequality , triangleInequalitySetup , "UseAreaCoordinates " → True]]}},
 Frame → All, Alignment → {{Left, Center , Center , Center }}]
```

Theorem	Area Coordinates	Time	Result
Ceva's Theorem	no	$3. \times 10^{-6}$	VerifyConjecture [(AreaMethod`SignedDistance (a, f) AreaMethod`SignedDistance (b, d) AreaMethod`SignedDistance ( c, e)) / (AreaMethod`SignedDistance (e, a) AreaMethod`SignedDistance ( f , b) AreaMethod`SignedDistance (d , c)) = 1, AreaMethod`IsIntersection [ f, c, p, a, b][ AreaMethod`IsIntersection [e, b, p, a, c][AreaMethod`IsIntersection [ d, a, p, b, c][ AreaMethod`FreePoint [a, b, c, p]]]]]
Desargues Theorem	no	$2. \times 10^{-6}$	VerifyConjecture [Parallel(b, c, b2, c2), AreaMethod`OnInterLineParallel [ c2, a2, x, c, a, c][ AreaMethod`OnInterLineParallel [b2, a2, x, b, a, b][AreaMethod`OnLine [x, a, a2][ AreaMethod`FreePoint [a, b, c, a2]]]]]

Out[128]=

Euler Line	yes	$3. \times 10^{-6}$	VerifyConjecture [AreaMethod`Collinear (x, y, z), AreaMethod`IsCentroid [z, a, b, c][AreaMethod`IsOrthocenter [ y, a, b, c][ AreaMethod`IsCircumcenter [x, a, b, c][AreaMethod`FreePoint [a, b, c]]]], UseAreaCoordinates → True]
Gauss-Newton Line	yes	$3. \times 10^{-6}$	VerifyConjecture [AreaMethod`Collinear (m1, m2, m3), AreaMethod`IsMidpoint [m3, x, y][AreaMethod`IsMidpoint [m2, a0, a2][AreaMethod`IsMidpoint [m1, a1, a3][AreaMethod`IsIntersection [ y, a0, a1, a2, a3][ AreaMethod`IsIntersection [x, a0, a3, a1, a2][AreaMethod`FreePoint [a0, a1, a2, a3]]]]]], UseAreaCoordinates → True]
Heron's Formula	yes	$1. \times 10^{-6}$	VerifyConjecture [16 AreaMethod`SignedArea ( a, b, c)$^2$ = AreaMethod`PythagoreanDifference(b, a, b) AreaMethod`PythagoreanDifference(a, c, a) - AreaMethod`PythagoreanDifference(c, a, b)^2, AreaMethod`FreePoint [a, b, c], UseAreaCoordinates → True]

Intercept Theorem	no	$2.\times 10^{-6}$	VerifyConjecture[AreaMethod`SignedDistance (s,a) / AreaMethod`SignedDistance (a,b) = AreaMethod`SignedDistance (s,c) / AreaMethod`SignedDistance (c,d), AreaMethod`IsIntersection [s, a, b, c, d][AreaMethod`OnParallel [d, b, a, c, r][AreaMethod`FreePoint [a, b, c]]], CheckParallelRatios → True]
Midpoint Theorem	no	$2.\times 10^{-6}$	VerifyConjecture [AreaMethod`Parallel (a, b, d, e), AreaMethod`IsMidpoint [e, a, c][AreaMethod`IsMidpoint [d, b, c][ AreaMethod`FreePoint [a, b, c]]]]
Menelaus' Theorem	no	0.	VerifyConjecture [(AreaMethod`SignedDistance (a, f) AreaMethod`SignedDistance (b, d) AreaMethod`SignedDistance ( c, e)) / (AreaMethod`SignedDistance (e, a) AreaMethod`SignedDistance ( f, b) AreaMethod`SignedDistance (d, c)) = -1, AreaMethod`IsIntersection [ f, x, y, a, b][ AreaMethod`IsIntersection [e, x, y, a, c][AreaMethod`IsIntersection [ d, x, y, b, c][ AreaMethod`FreePoint [a, b, c, x, y]]]]]

Pappus's Line Theorem	no	$3. \times 10^{-6}$	VerifyConjecture[AreaMethod`Collinear(x, y, z), AreaMethod`IsIntersection[z, B1, C2, B2, C1][AreaMethod`IsIntersection[ y, A1, C2, A2, C1][ AreaMethod`IsIntersection[x, A1, B2, A2, B1][AreaMethod`OnLine[C2, A2, B2][ AreaMethod`OnLine[C1, A1, B1][AreaMethod`FreePoint[A1, A2, B1, B2]]]]]]
Pythagorean Theorem	no	$3. \times 10^{-6}$	VerifyConjecture[AreaMethod`PythagoreanDifference[c, a, b] == 0, AreaMethod`OnPerp[c, a, b][AreaMethod`FreePoint[a, b]]]
Triangle Inequality	yes	$3. \times 10^{-6}$	VerifyConjecture[√AreaMethod`PythagoreanDifference[a,b,a] / √2 + √AreaMethod`PythagoreanDifference[b,c,b] / √2 ≥ $\frac{1}{2}$ √AreaMethod`PythagoreanDifference[c, a, c], AreaMethod`ECS1[a, b, c], UseAreaCoordinates → True]

Further Work

- Extend to higher dimensions.
- Work in different geometries (Poincaré disc seems like a good candidate).
- Integrate with existing synthetic geometry functionality.