International Journal of Computer Discovered Mathematics (IJCDM)
ISSN 2367-7775 ©IJCDM
Volume 9, 2024, pp. 242-248.
Received July 14, 2024. Published on-line December 31, 2024.
web: http://www.journal-1.eu/
©The Author(s) This article is published with open access[1].

Generalization of Schiffler's Theorem

Benjamin Lee Warren[a] [2]
University of Binghamton, Mathematics,
4400 Vestel Pkwy E, Binghamton, NY 13902
e-mail: warrenbenjamin52@yahoo.com
web: https://orcid.org/0000-0002-2829-6953

Mathematics Subject Classification (2010). 51N20.

1. Introduction

Schiffler's theorem [1] is a recent result in triangle geometry from the 1980's. It states that if one partitions a triangle into three others by the incenter and the vertices, then the Euler lines of those three triangles and the original are concurrent. Here, we give a generalization of this result using ideas from [2] and [3].

2. Result

Theorem 2.1. *Let A, B, and C denote the vertices of a given triangle.*
Let D be the incenter of A, B, and C.
Let E be a point on line AD.
Let F be the point on line BD such that $BF : DF = AE : DE$.
Let G be the point on line CD such that $CG : DG = AE : DE$.
Let H be the midpoint of E and F.
Let I be the midpoint of E and G.
Let J be the midpoint of F and G.
Let K be the midpoint of D and E.
Let L be the midpoint of D and F.
Let M be the midpoint of D and G.
Let lines AL, BK, and DH intersect at point N.
Let lines BM, CL, and DJ intersect at point O.
Let lines AM, CK, and DI intersect at point P.

[1]This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

[2]Corresponding author

Let Q be the circumcenter of $A, B,$ and C.
Let R be the nine point center of $E, F,$ and G.
Let S be the circumcenter of $A, B,$ and D.
Let T be the circumcenter of $B, C,$ and D.
Let U be the circumcenter of $A, C,$ and D.
Let V be the nine point center of $D, E,$ and F.
Let W be the nine point center of $D, F,$ and G.
Let Z be the nine point center of $D, E,$ and G.
Then lines $NSV, OTW, PUZ,$ and QR are concurrent.

Proof. Without loss of generality, we will work in Cartesian coordinates.
A has coordinates $(0, 0)$.
B has coordinates $(1, 0)$.
C has coordinates (x_C, y_C).
D has coordinates $(\frac{x_C+\sqrt{x_C^2+y_C^2}}{1+\sqrt{x_C^2+y_C^2}+\sqrt{(x_C-1)^2+y_C^2}}, \frac{y_C}{1+\sqrt{x_C^2+y_C^2}+\sqrt{(x_C-1)^2+y_C^2}})$.
E has coordinates $(x_E, \frac{x_E y_D}{x_D})$.
F has coordinates $(1 + \frac{x_E(x_D-1)}{x_D}, \frac{x_E y_D}{x_D})$.
G has coordinates $(x_C + \frac{x_E(x_D-x_C)}{x_D}, y_C + \frac{x_E(y_D-y_C)}{x_D})$.
H has coordinates $(\frac{1}{2}(x_E + 1 + \frac{x_E(x_D-1)}{x_D}), \frac{x_E y_D}{x_D})$.
I has coordinates $(\frac{1}{2}(x_E + x_C + \frac{x_E(x_D-x_C)}{x_D}), \frac{2x_E y_D+y_C(x_D-x_E)}{2x_D})$.
J has coordinates $(\frac{x_D(x_C+1)+x_E(2x_D-x_C-1)}{2x_D}, \frac{2x_E y_D+y_C(x_D-x_E)}{2x_D})$.
K has coordinates $(\frac{1}{2}(x_D + x_E), \frac{1}{2}(y_D + \frac{x_E y_D}{x_D}))$.
L has coordinates $(\frac{1}{2}(x_D + 1 + \frac{x_E(x_D-1)}{x_D}), \frac{1}{2}(y_D + \frac{x_E y_D}{x_D}))$.
M has coordinates $(\frac{1}{2}(x_D + x_C + \frac{x_E(x_D-x_C)}{x_D}), \frac{1}{2}(y_D + y_C + \frac{x_E(y_D-y_C)}{x_D}))$.
N has coordinates $(\frac{-x_D^2-x_D x_E-x_D+x_E}{x_E-3x_D}, \frac{y_D(x_D+x_E)}{3x_D-x_E})$.
O has coordinates $(\frac{x_D-x_E+x_C x_D-x_C x_E+x_D x_E+x_D^2}{3x_D-x_E}, \frac{x_D y_C+x_D y_D-x_E y_C+x_E y_D}{3x_D-x_E})$.
P has coordinates $(\frac{-x_C x_D+x_C x_E-x_D^2-x_D x_E}{x_E-3x_D}, \frac{-x_D y_C-x_D y_D+x_E y_C-x_E y_D}{x_E-3x_D})$.
Q has coordinates $(\frac{1}{2}, \frac{x_C(x_C-1)+y_C^2}{2y_C})$.
R has coordinates $(\frac{x_D(2x_C+1)+x_E(4x_D-2x_C-1)}{4x_D}, \frac{4x_E y_C y_D+y_C^2(x_D-x_E)+x_E(x_C^2-x_C)+x_D(x_C-x_C^2)}{4x_D y_C})$.
S has coordinates $(\frac{1}{2}, \frac{x_D(x_D-1)+y_D^2}{2y_D})$.
T has coordinates $(\frac{x_C^2 y_D-x_D^2 y_C+y_C^2 y_D-y_C y_D^2+y_C-y_D}{2(x_C y_D-x_D y_C+y_C-y_D)}, \frac{-x_C^2 x_D+x_C^2+x_C x_D^2+x_C y_D^2-x_C-x_D^2-x_D y_C^2+x_D+y_C^2-y_D^2}{2(x_C y_D-x_D y_C+y_C-y_D)})$.
U has coordinates $(\frac{x_C^2 y_D-x_D^2 y_C+y_C^2 y_D-y_C y_D^2}{2(x_C y_D-x_D y_C)}, \frac{-x_C^2 x_D+x_C x_D^2+x_C y_D^2-x_D y_C^2}{2(x_C y_D-x_D y_C)})$.
V has coordinates $(\frac{x_D(2x_D+1)+x_E(2x_D-1)}{4x_D}, \frac{y_D^2(3x_E+x_D)+x_D x_E(x_D-1)-x_D^3+x_D^2}{4x_D y_D})$.
W has coordinates
$(-((x_E - x_D)y_C y_D^2 - (x_E y_C^2 - x_D y_C^2 + 2x_C x_D x_E - 2x_D x_E - x_C^2 x_E + x_E + (x_C - 1)x_D(2x_D+x_C+1))y_D+(x_D-1)((3x_D-2x_C-1)x_E+x_D(x_D+2x_C+1))y_C)/(4x_D((x_C-1)y_D-(x_D-1)y_C)), ((x_C-1)(3x_E+x_D)y_D^2-2((x_D+x_C-2)x_E+x_D(x_D-x_C))y_C y_D+(x_D-1)(x_E-x_D)y_C^2+(x_C-1)(x_D-1)(x_D-x_C)x_E-(x_C-1)x_D^3+(x_C-1)(x_C+1)x_D^2-(x_C-1)x_C x_D)/(4x_D((x_C-1)y_D-(x_D-1)y_C)))$.
Z has coordinates
$(-((x_E - x_D)y_C y_D^2 + ((x_D - x_E)y_C^2 + (x_C^2 - 2x_C x_D)x_E - 2x_C x_D^2 - x_C^2 x_D)y_D + ((3x_D^2 - 2x_C x_D)x_E + x_D^3 + 2x_C x_D^2)y_C)/(4x_D(x_C y_D - x_D y_C)), (x_C(3x_E + x_D)y_D^2 -$

$2(x_Dx_E+x_Cx_E+x_D^2-x_Cx_D)y_Cy_D+x_D(x_E-x_D)y_C^2+x_Cx_D(x_D-x_C)x_E-x_Cx_D^3+x_C^2x_D^2)/(4x_D(x_Cy_D-x_Dy_C)))$.
Line NSV is given by the equation
$y=(((3x_E-x_D)y_D^2+(x_D-1)x_Dx_E-3x_D^3+3x_D^2)x+((x_D-2)x_E+x_D^2)y_D^2+(x_D-1)^2x_Dx_E+x_D^4-x_D^2)/((2x_D-1)(x_E+x_D)y_D)$
$\iff$
$x(\frac{y_N-y_S}{x_N-x_S})-y+y_S-x_S(\frac{y_N-y_S}{x_N-x_S})=0.$
Line OTW is given by the equation
$y=(((x_Ey_D+x_Dy_D-x_Ey_C+x_Dy_C)/(3x_D-x_E)-(x_Cy_D^2-y_D^2-x_Dy_C^2+y_C^2+x_Cx_D^2-x_D^2-x_C^2x_D+x_D+x_C^2-x_C)/(2(x_Cy_D-y_D-x_Dy_C+y_C)))(x-(x_Dx_E-x_Cx_E-x_E+x_D^2+x_Cx_D+x_D)/(3x_D-x_E)))/((x_Dx_E-x_Cx_E-x_E+x_D^2+x_Cx_D+x_D)/(3x_D-x_E)-(-y_Cy_D^2+y_C^2y_D+x_C^2y_D-y_D-x_D^2y_C+y_C)/(2(x_Cy_D-y_D-x_Dy_C+y_C)))+(x_Ey_D+x_Dy_D-x_Ey_C+x_Dy_C)/(3x_D-x_E)$
$\iff$
$x(\frac{y_O-y_T}{x_O-x_T})-y+y_T-x_T(\frac{y_O-y_T}{x_O-x_T})=0.$
Line PUZ is given by the equation
$y=(((-x_Ey_D-x_Dy_D+x_Ey_C-x_Dy_C)/(x_E-3x_D)-(x_Cy_D^2-x_Dy_C^2+x_Cx_D^2-x_C^2x_D)/(2(x_Cy_D-x_Dy_C)))(x-(-x_Dx_E+x_Cx_E-x_D^2-x_Cx_D)/(x_E-3x_D)))/((-x_Dx_E+x_Cx_E-x_D^2-x_Cx_D)/(x_E-3x_D)-(-y_Cy_D^2+y_C^2y_D+x_C^2y_D-x_D^2y_C)/(2(x_Cy_D-x_Dy_C)))+(-x_Ey_D-x_Dy_D+x_Ey_C-x_Dy_C)/(x_E-3x_D)$
$\iff$
$x(\frac{y_P-y_U}{x_P-x_U})-y+y_U-x_U(\frac{y_P-y_U}{x_P-x_U})=0.$
Line QR is given by the equation
$y=(((4x_Ey_Cy_D+(x_D-x_E)y_C^2+(x_C^2-x_C)x_E+(x_C-x_C^2)x_D)/(4x_Dy_C)-(y_C^2+(x_C-1)x_C)/(2y_C))(x-1/2))/(((4x_D-2x_C-1)x_E+(2x_C+1)x_D)/(4x_D)-1/2)+(y_C^2+(x_C-1)x_C)/(2y_C)$
$\iff$
$x(\frac{y_Q-y_R}{x_Q-x_R})-y+y_R-x_R(\frac{y_Q-y_R}{x_Q-x_R})=0.$
We evaluate the determinants

$$det\begin{bmatrix}\frac{y_N-y_S}{x_N-x_S} & -1 & y_S-x_S(\frac{y_N-y_S}{x_N-x_S})\\ \frac{y_O-y_T}{x_O-x_T} & -1 & y_T-x_T(\frac{y_O-y_T}{x_O-x_T})\\ \frac{y_P-y_U}{x_P-x_U} & -1 & y_U-x_U(\frac{y_P-y_U}{x_P-x_U})\end{bmatrix}$$

and

$$det\begin{bmatrix}\frac{y_N-y_S}{x_N-x_S} & -1 & y_S-x_S(\frac{y_N-y_S}{x_N-x_S})\\ \frac{y_O-y_T}{x_O-x_T} & -1 & y_T-x_T(\frac{y_O-y_T}{x_O-x_T})\\ \frac{y_Q-y_R}{x_Q-x_R} & -1 & y_R-x_R(\frac{y_Q-y_R}{x_Q-x_R})\end{bmatrix}.$$

The determinant of first matrix is
$(-3*x_C^5*x_D^4*y_D+4*x_C^5*x_D^3*x_E*y_D+3*x_C^5*x_D^3*y_D-x_C^5*x_D^2*x_E^2*y_D-4*x_C^5*x_D^2*x_E*y_D-x_C^5*x_D^2*y_D^3+x_C^5*x_D*x_E^2*y_D+4*x_C^5*x_D*x_E*y_D^3-3*x_C^5*x_E^2*y_D^3+3*x_C^4*x_D^5*y_D-4*x_C^4*x_D^4*x_E*y_D+3*x_C^4*x_D^4*y_D+x_C^4*x_D^3*x_E^2*y_D-4*x_C^4*x_D^3*x_E*y_D-2*x_C^4*x_D^3*y_C*y_D^2+3*x_C^4*x_D^3*y_D^3-6*x_C^4*x_D^3*y_D+x_C^4*x_D^2*x_E^2*y_D-12*x_C^4*x_D^2*x_E*y_D^3+8*x_C^4*x_D^2*x_E*y_D+x_C^4*x_D^2*y_C*y_D^2+x_C^4*x_D^2*y_D^3+2*x_C^4*x_D*x_E^2*y_C*y_D^2+9*x_C^4*x_D*x_E^2*y_D^3-2*x_C^4*x_D*x_E^2*y_D-4*x_C^4*x_D*x_E*y_D^3-x_C^4*x_E^2*y_C*y_D^2+3*x_C^4*x_E^2*y_D^3+3*x_C^3*x_D^6*y_C-4*x_C^3*x_D^5*x_E*y_C-6*x_C^3*x_D^5*y_C-6*x_C^3*x_D^5*y_D+x_C^3*x_D^4*x_E^2*y_C+8*x_C^3*x_D^4*x_E*y_C+8*x_C^3*x_D^4*x_E*y_D-6*x_C^3*x_D^4*y_C^2*y_D+6*x_C^3*x_D^4*y_C*y_D^2+3*x_C^3*x_D^4*y_C+3*x_C^3*x_D^4*y_D-2*x_C^3*x_D^3*x_E^2*y_C-2*x_C^3*x_D^3*x_E^2*y_D+8*x_C^3*x_D^3*x_E*y_C^2*y_D-4*x_C^3*x_D^3*x_E*y_C-4*x_C^3*x_D^3*x_E*y_D+6*x_C^3*x_D^3*y_C^2*y_D-2*x_C^3*x_D^3*y_C*y_D^2-6*x_C^3*x_D^3*y_D^3+3*x_C^3*x_D^3*y_D-2*x_C^3*x_D^2*x_E^2*y_C^2*y_D-6*x_C^3*x_D^2*x_E^2*y_C*y_D^2+x_C^3*x_D^2*x_E^2*y_C+$

$x_C^3*x_D^2*x_E^2*y_D-8*x_C^3*x_D^2*x_E*y_C^2*y_D+24*x_C^3*x_D^2*x_E*y_D^3-4*x_C^3*x_D^2*x_E*y_D-2*$
$x_C^3*x_D^2*y_C^2*y_D^3+3*x_C^3*x_D^2*y_C*y_D^4-3*x_C^3*x_D^2*y_C*y_D^2+x_C^3*x_D^2*y_D^3+2*x_C^3*x_D*x_E^2*$
$y_C^2*y_D+2*x_C^3*x_D*x_E^2*y_C*y_D^2-18*x_C^3*x_D*x_E^2*y_D^3+x_C^3*x_D*x_E^2*y_D+8*x_C^3*x_D*x_E*$
$y_C^2*y_D^3-12*x_C^3*x_D*x_E*y_C*y_D^4+4*x_C^3*x_D*x_E*y_C*y_D^2-4*x_C^3*x_D*x_E*y_D^3-6*x_C^3*$
$x_E^2*y_C^2*y_D^3+9*x_C^3*x_E^2*y_C*y_D^4-x_C^3*x_E^2*y_C*y_D^2+3*x_C^3*x_E^2*y_D^3-3*x_C^2*x_D^7*y_C+4*$
$x_C^2*x_D^6*x_E*y_C+3*x_C^2*x_D^6*y_C-x_C^2*x_D^5*x_E^2*y_C-4*x_C^2*x_D^5*x_E*y_C+6*x_C^2*x_D^5*y_C^2*$
$y_D-6*x_C^2*x_D^5*y_C*y_D^2+3*x_C^2*x_D^5*y_C+3*x_C^2*x_D^5*y_D+x_C^2*x_D^4*x_E^2*y_C-8*x_C^2*x_D^4*$
$x_E*y_C^2*y_D+16*x_C^2*x_D^4*x_E*y_C*y_D^2-4*x_C^2*x_D^4*x_E*y_C-4*x_C^2*x_D^4*x_E*y_D-3*x_C^2*$
$x_D^4*y_C-3*x_C^2*x_D^4*y_D+2*x_C^2*x_D^3*x_E^2*y_C^2*y_D-10*x_C^2*x_D^3*x_E^2*y_C*y_D^2+x_C^2*x_D^3*x_E^2*$
$y_C+x_C^2*x_D^3*x_E^2*y_D-24*x_C^2*x_D^3*x_E*y_C*y_D^2+4*x_C^2*x_D^3*x_E*y_C+4*x_C^2*x_D^3*x_E*y_D-$
$4*x_C^2*x_D^3*y_C^3*y_D^2+6*x_C^2*x_D^3*y_C^2*y_D^3-6*x_C^2*x_D^3*y_C^2*y_D-3*x_C^2*x_D^3*y_C*y_D^4+5*x_C^2*$
$x_D^3*y_C*y_D^2+3*x_C^2*x_D^3*y_D^3+24*x_C^2*x_D^2*x_E^2*y_C*y_D^2-x_C^2*x_D^2*x_E^2*y_C-x_C^2*x_D^2*x_E^2*$
$y_D-8*x_C^2*x_D^2*x_E*y_C^2*y_D^3+8*x_C^2*x_D^2*x_E*y_C^2*y_D+12*x_C^2*x_D^2*x_E*y_C*y_D^4+4*x_C^2*$
$x_D^2*x_E*y_C*y_D^2-12*x_C^2*x_D^2*x_E*y_D^3+2*x_C^2*x_D^2*y_C^3*y_D^2-3*x_C^2*x_D^2*y_C*y_D^4+2*x_C^2*$
$x_D^2*y_C*y_D^2-x_C^2*x_D^2*y_D^3+4*x_C^2*x_D*x_E^2*y_C^3*y_D^2+2*x_C^2*x_D*x_E^2*y_C^2*y_D^3-2*x_C^2*x_D*$
$x_E^2*y_C^2*y_D-9*x_C^2*x_D*x_E^2*y_C*y_D^4-9*x_C^2*x_D*x_E^2*y_C*y_D^2+9*x_C^2*x_D*x_E^2*y_D^3-8*$
$x_C^2*x_D*x_E*y_C^2*y_D^3+12*x_C^2*x_D*x_E*y_C*y_D^4-4*x_C^2*x_D*x_E*y_C*y_D^2+4*x_C^2*x_D*x_E*$
$y_D^3-2*x_C^2*x_E^2*y_C^3*y_D^2+8*x_C^2*x_E^2*y_C^2*y_D^3-9*x_C^2*x_E^2*y_C*y_D^4+2*x_C^2*x_E^2*y_C*y_D^2-3*$
$x_C^2*x_E^2*y_D^3+3*x_C*x_D^7*y_C-4*x_C*x_D^6*x_E*y_C+3*x_C*x_D^6*y_C^3-2*x_C*x_D^6*y_C^2*y_D-6*$
$x_C*x_D^6*y_C+x_C*x_D^5*x_E^2*y_C-4*x_C*x_D^5*x_E*y_C^3-8*x_C*x_D^5*x_E*y_C^2*y_D+8*x_C*x_D^5*$
$x_E*y_C-6*x_C*x_D^5*y_C^3-2*x_C*x_D^5*y_C^2*y_D+6*x_C*x_D^5*y_C*y_D^2+3*x_C*x_D^5*y_C+x_C*$
$x_D^4*x_E^2*y_C^3+10*x_C*x_D^4*x_E^2*y_C^2*y_D-2*x_C*x_D^4*x_E^2*y_C+8*x_C*x_D^4*x_E*y_C^3+24*x_C*$
$x_D^4*x_E*y_C^2*y_D-16*x_C*x_D^4*x_E*y_C*y_D^2-4*x_C*x_D^4*x_E*y_C-3*x_C*x_D^4*y_C^4*y_D+6*$
$x_C*x_D^4*y_C^3*y_D^2+3*x_C*x_D^4*y_C^3-4*x_C*x_D^4*y_C^2*y_D^3+5*x_C*x_D^4*y_C^2*y_D-6*x_C*x_D^4*y_C*$
$y_D^2-2*x_C*x_D^3*x_E^2*y_C^3-22*x_C*x_D^3*x_E^2*y_C^2*y_D+10*x_C*x_D^3*x_E^2*y_C*y_D^2+x_C*x_D^3*$
$x_E^2*y_C+4*x_C*x_D^3*x_E*y_C^4*y_D-4*x_C*x_D^3*x_E*y_C^3-20*x_C*x_D^3*x_E*y_C^2*y_D+24*x_C*$
$x_D^3*x_E*y_C*y_D^2+3*x_C*x_D^3*y_C^4*y_D-2*x_C*x_D^3*y_C^3*y_D^2-2*x_C*x_D^3*y_C^2*y_D^3-x_C*x_D^3*$
$y_C^2*y_D+3*x_C*x_D^3*y_C*y_D^4-x_C*x_D^3*y_C*y_D^2-x_C*x_D^2*x_E^2*y_C^4*y_D-6*x_C*x_D^2*x_E^2*y_C^3*$
$y_D^2+x_C*x_D^2*x_E^2*y_C^3+4*x_C*x_D^2*x_E^2*y_C^2*y_D^3+15*x_C*x_D^2*x_E^2*y_C^2*y_D-18*x_C*x_D^2*$
$x_E^2*y_C*y_D^2-4*x_C*x_D^2*x_E*y_C^4*y_D+8*x_C*x_D^2*x_E*y_C^2*y_D^3+4*x_C*x_D^2*x_E*y_C^2*y_D-$
$12*x_C*x_D^2*x_E*y_C*y_D^4-4*x_C*x_D^2*x_E*y_C*y_D^2-x_C*x_D^2*y_C^4*y_D^3+3*x_C*x_D^2*y_C^3*y_D^4-$
$3*x_C*x_D^2*y_C^3*y_D^2-2*x_C*x_D^2*y_C^2*y_D^5+3*x_C*x_D^2*y_C^2*y_D^3+x_C*x_D*x_E^2*y_C^4*y_D+2*x_C*$
$x_D*x_E^2*y_C^3*y_D^2-6*x_C*x_D*x_E^2*y_C^2*y_D^3-3*x_C*x_D*x_E^2*y_C^2*y_D+9*x_C*x_D*x_E^2*y_C*$
$y_D^4+5*x_C*x_D*x_E^2*y_C*y_D^2+4*x_C*x_D*x_E*y_C^4*y_D^3-12*x_C*x_D*x_E*y_C^3*y_D^4+4*x_C*$
$x_D*x_E*y_C^3*y_D^2+8*x_C*x_D*x_E*y_C^2*y_D^5-4*x_C*x_D*x_E*y_C^2*y_D^3-3*x_C*x_E^2*y_C^4*y_D^3+$
$9*x_C*x_E^2*y_C^3*y_D^4-x_C*x_E^2*y_C^3*y_D^2-6*x_C*x_E^2*y_C^2*y_D^5+x_C*x_E^2*y_C^2*y_D^3-x_D^7*y_C^3+4*$
$x_D^6*x_E*y_C^3+x_D^6*y_C^3+x_D^6*y_C^2*y_D-3*x_D^5*x_E^2*y_C^3-8*x_D^5*x_E*y_C^3+4*x_D^5*x_E*y_C^2*y_D+$
$3*x_D^5*y_C^4*y_D-2*x_D^5*y_C^3*y_D^2+x_D^5*y_C^3-3*x_D^5*y_C^2*y_D+7*x_D^4*x_E^2*y_C^3-5*x_D^4*x_E^2*y_C^2*$
$y_D-4*x_D^4*x_E*y_C^4*y_D+4*x_D^4*x_E*y_C^3-4*x_D^4*x_E*y_C^2*y_D-3*x_D^4*y_C^4*y_D-x_D^4*y_C^3+$
$2*x_D^4*y_C^2*y_D^3+2*x_D^4*y_C^2*y_D+x_D^3*x_E^2*y_C^4*y_D+2*x_D^3*x_E^2*y_C^3*y_D^2-5*x_D^3*x_E^2*y_C^3+$
$7*x_D^3*x_E^2*y_C^2*y_D+4*x_D^3*x_E*y_C^4*y_D-2*x_D^3*y_C^5*y_D^2+3*x_D^3*y_C^4*y_D^3-x_D^3*y_C^3*y_D^4+$
$3*x_D^3*y_C^3*y_D^2-3*x_D^3*y_C^2*y_D^3-x_D^2*x_E^2*y_C^4*y_D+x_D^2*x_E^2*y_C^3-2*x_D^2*x_E^2*y_C^2*y_D^3-2*$
$x_D^2*x_E^2*y_C^2*y_D+4*x_D^2*x_E*y_C^4*y_D^3-4*x_D^2*x_E*y_C^3*y_D^4-4*x_D^2*x_E*y_C^3*y_D^2+4*x_D^2*$
$x_E*y_C^2*y_D^3+x_D^2*y_C^5*y_D^2-x_D^2*y_C^4*y_D^3-x_D^2*y_C^3*y_D^4+x_D^2*y_C^2*y_D^5+2*x_D*x_E^2*y_C^5*y_D^2-$
$7*x_D*x_E^2*y_C^4*y_D^3+5*x_D*x_E^2*y_C^3*y_D^4+x_D*x_E^2*y_C^3*y_D^2-x_D*x_E^2*y_C^2*y_D^3-4*x_D*x_E*$
$y_C^4*y_D^3+8*x_D*x_E*y_C^3*y_D^4-4*x_D*x_E*y_C^2*y_D^5-x_E^2*y_C^5*y_D^2+5*x_E^2*y_C^4*y_D^3-7*x_E^2*y_C^3*$
$y_D^4+3*x_E^2*y_C^2*y_D^5)/(y_D*(2*x_D-1)*(-x_C^2*x_D*y_D-x_E*x_C^2*y_D-2*x_C*x_D^2*y_C+2*$
$x_C*x_D^2*y_D+2*x_E*x_C*x_D*y_C+2*x_E*x_C*x_D*y_D+x_D^3*y_C-3*x_E*x_D^2*y_C-3*x_D*$
$y_C^2*y_D+3*x_D*y_C*y_D^2+x_E*y_C^2*y_D-x_E*y_C*y_D^2)*(x_D*y_C-x_D*y_D+x_E*y_C-x_E*y_D-$

$x_D^3*y_C+2*x_D^2*y_D+2*x_C*x_D^2*y_C-2*x_C*x_D^2*y_D+x_C^2*x_D*y_D+x_C^2*x_E*y_D+3*x_D^2*x_E*y_C-3*x_D*y_C*y_D^2+3*x_D*y_C^2*y_D+x_E*y_C*y_D^2-x_E*y_C^2*y_D-2*x_C*x_D*y_C+2*x_C*x_E*y_C-4*x_D*x_E*y_C+2*x_D*x_E*y_D-2*x_C*x_D*x_E*y_C-2*x_C*x_D*x_E*y_D))$.

The determinant of second matrix is

$(-3*x_C^5*x_D^4*y_D+4*x_C^5*x_D^3*x_E*y_D+3*x_C^5*x_D^3*y_D-x_C^5*x_D^2*x_E^2*y_D-4*x_C^5*x_D^2*x_E*y_D-x_C^5*x_D^2*y_D^3+x_C^5*x_D*x_E^2*y_D+4*x_C^5*x_D*x_E*y_D^3-3*x_C^5*x_E^2*y_D^3+3*x_C^4*x_D^5*y_D-4*x_C^4*x_D^4*x_E*y_D+3*x_C^4*x_D^4*y_D+x_C^4*x_D^3*x_E^2*y_D-4*x_C^4*x_D^3*x_E*y_D-2*x_C^4*x_D^3*y_C*y_D^2+3*x_C^4*x_D^3*y_D^3-6*x_C^4*x_D^3*y_D+x_C^4*x_D^2*x_E^2*y_D-12*x_C^4*x_D^2*x_E*y_D^3+8*x_C^4*x_D^2*x_E*y_D+x_C^4*x_D^2*y_C*y_D^2+x_C^4*x_D^2*y_D^3+2*x_C^4*x_D*x_E^2*y_C*y_D^2+9*x_C^4*x_D*x_E^2*y_D^3-2*x_C^4*x_D*x_E^2*y_D-4*x_C^4*x_D*x_E*y_D^3-x_C^4*x_E^2*y_C*y_D^2+3*x_C^4*x_E^2*y_D^3+3*x_C^3*x_D^6*y_C-4*x_C^3*x_D^5*x_E*y_C-6*x_C^3*x_D^5*y_C-6*x_C^3*x_D^5*y_D+x_C^3*x_D^4*x_E^2*y_C+8*x_C^3*x_D^4*x_E*y_C+8*x_C^3*x_D^4*x_E*y_D-6*x_C^3*x_D^4*y_C^2*y_D+6*x_C^3*x_D^4*y_C*y_D^2+3*x_C^3*x_D^4*y_C+3*x_C^3*x_D^4*y_D-2*x_C^3*x_D^3*x_E^2*y_C-2*x_C^3*x_D^3*x_E^2*y_D+8*x_C^3*x_D^3*x_E*y_C^2*y_D-4*x_C^3*x_D^3*x_E*y_C-4*x_C^3*x_D^3*x_E*y_D+6*x_C^3*x_D^3*y_C^2*y_D-2*x_C^3*x_D^3*y_C*y_D^2-6*x_C^3*x_D^3*y_D^3+3*x_C^3*x_D^3*y_D-2*x_C^3*x_D^2*x_E^2*y_C^2*y_D-6*x_C^3*x_D^2*x_E^2*y_C*y_D^2+x_C^3*x_D^2*x_E^2*y_C+x_C^3*x_D^2*x_E^2*y_D-8*x_C^3*x_D^2*x_E*y_C^2*y_D+24*x_C^3*x_D^2*x_E*y_D^3-4*x_C^3*x_D^2*x_E*y_D-2*x_C^3*x_D^2*y_C^2*y_D^3+3*x_C^3*x_D^2*y_C*y_D^4-3*x_C^3*x_D^2*y_C*y_D^2+x_C^3*x_D^2*y_D^3+2*x_C^3*x_D*x_E^2*y_C^2*y_D+2*x_C^3*x_D*x_E^2*y_C*y_D^2-18*x_C^3*x_D*x_E^2*y_D^3+x_C^3*x_D*x_E^2*y_D+8*x_C^3*x_D*x_E*y_C^2*y_D^3-12*x_C^3*x_D*x_E*y_C*y_D^4+4*x_C^3*x_D*x_E*y_C*y_D^2-4*x_C^3*x_D*x_E*y_D^3-6*x_C^3*x_E^2*y_C^2*y_D^3+9*x_C^3*x_E^2*y_C*y_D^4-x_C^3*x_E^2*y_C*y_D^2+3*x_C^3*x_E^2*y_D^3-3*x_C^2*x_D^7*y_C+4*x_C^2*x_D^6*x_E*y_C+3*x_C^2*x_D^6*y_C-x_C^2*x_D^5*x_E^2*y_C-4*x_C^2*x_D^5*x_E*y_C+6*x_C^2*x_D^5*y_C^2*y_D-6*x_C^2*x_D^5*y_C*y_D^2+3*x_C^2*x_D^5*y_C+3*x_C^2*x_D^5*y_D+x_C^2*x_D^4*x_E^2*y_C-8*x_C^2*x_D^4*x_E*y_C^2*y_D+16*x_C^2*x_D^4*x_E*y_C*y_D^2-4*x_C^2*x_D^4*x_E*y_C-4*x_C^2*x_D^4*x_E*y_D-3*x_C^2*x_D^4*y_C-3*x_C^2*x_D^4*y_D+2*x_C^2*x_D^3*x_E^2*y_C^2*y_D-10*x_C^2*x_D^3*x_E^2*y_C*y_D^2+x_C^2*x_D^3*x_E^2*y_C+x_C^2*x_D^3*x_E^2*y_D-24*x_C^2*x_D^3*x_E*y_C*y_D^2+4*x_C^2*x_D^3*x_E*y_C+4*x_C^2*x_D^3*x_E*y_D-4*x_C^2*x_D^3*y_C^3*y_D^2+6*x_C^2*x_D^3*y_C^2*y_D^3-6*x_C^2*x_D^3*y_C^2*y_D-3*x_C^2*x_D^3*y_C*y_D^4+5*x_C^2*x_D^3*y_C*y_D^2+3*x_C^2*x_D^3*y_D^3+24*x_C^2*x_D^2*x_E^2*y_C*y_D^2-x_C^2*x_D^2*x_E^2*y_C-x_C^2*x_D^2*x_E^2*y_D-8*x_C^2*x_D^2*x_E*y_C^2*y_D^3+8*x_C^2*x_D^2*x_E*y_C^2*y_D+12*x_C^2*x_D^2*x_E*y_C*y_D^4+4*x_C^2*x_D^2*x_E*y_C*y_D^2-12*x_C^2*x_D^2*x_E*y_D^3+2*x_C^2*x_D^2*y_C^3*y_D^2-3*x_C^2*x_D^2*y_C*y_D^4+2*x_C^2*x_D^2*y_C*y_D^2-x_C^2*x_D^2*y_D^3+4*x_C^2*x_D*x_E^2*y_C^3*y_D^2+2*x_C^2*x_D*x_E^2*y_C^2*y_D^3-2*x_C^2*x_D*x_E^2*y_C^2*y_D-9*x_C^2*x_D*x_E^2*y_C*y_D^4-9*x_C^2*x_D*x_E^2*y_C*y_D^2+9*x_C^2*x_D*x_E^2*y_D^3-8*x_C^2*x_D*x_E*y_C^2*y_D^3+12*x_C^2*x_D*x_E*y_C*y_D^4-4*x_C^2*x_D*x_E*y_C*y_D^2+4*x_C^2*x_D*x_E*y_D^3-2*x_C^2*x_E^2*y_C^3*y_D^2+8*x_C^2*x_E^2*y_C^2*y_D^3-9*x_C^2*x_E^2*y_C*y_D^4+2*x_C^2*x_E^2*y_C*y_D^2-3*x_C^2*x_E^2*y_D^3+3*x_C*x_D^7*y_C-4*x_C*x_D^6*x_E*y_C+3*x_C*x_D^6*y_C^3-2*x_C*x_D^6*y_C^2*y_D-6*x_C*x_D^6*y_C+x_C*x_D^5*x_E^2*y_C-4*x_C*x_D^5*x_E*y_C^3-8*x_C*x_D^5*x_E*y_C^2*y_D+8*x_C*x_D^5*x_E*y_C-6*x_C*x_D^5*y_C^3-2*x_C*x_D^5*y_C^2*y_D+6*x_C*x_D^5*y_C*y_D^2+3*x_C*x_D^5*y_C+x_C*x_D^4*x_E^2*y_C^3+10*x_C*x_D^4*x_E^2*y_C^2*y_D-2*x_C*x_D^4*x_E^2*y_C+8*x_C*x_D^4*x_E*y_C^3+24*x_C*x_D^4*x_E*y_C^2*y_D-16*x_C*x_D^4*x_E*y_C*y_D^2-4*x_C*x_D^4*x_E*y_C-3*x_C*x_D^4*y_C^4*y_D+6*x_C*x_D^4*y_C^3*y_D^2+3*x_C*x_D^4*y_C^3-4*x_C*x_D^4*y_C^2*y_D^3+5*x_C*x_D^4*y_C^2*y_D-6*x_C*x_D^4*y_C*y_D^2-2*x_C*x_D^3*x_E^2*y_C^3-22*x_C*x_D^3*x_E^2*y_C^2*y_D+10*x_C*x_D^3*x_E^2*y_C*y_D^2+x_C*x_D^3*x_E^2*y_C+4*x_C*x_D^3*x_E*y_C^4*y_D-4*x_C*x_D^3*x_E*y_C^3-20*x_C*x_D^3*x_E*y_C^2*y_D+24*x_C*x_D^3*x_E*y_C*y_D^2+3*x_C*x_D^3*y_C^4*y_D-2*x_C*x_D^3*y_C^3*y_D^2-2*x_C*x_D^3*y_C^2*y_D^3-x_C*x_D^3*y_C^2*y_D+3*x_C*x_D^3*y_C*y_D^4-x_C*x_D^3*y_C*y_D^2-x_C*x_D^2*x_E^2*y_C^4*y_D-6*x_C*x_D^2*x_E^2*y_C^3*y_D^2+x_C*x_D^2*x_E^2*y_C^3+4*x_C*x_D^2*x_E^2*y_C^2*y_D^3+15*x_C*x_D^2*x_E^2*y_C^2*y_D-18*x_C*x_D^2*x_E^2*y_C*y_D^2-4*x_C*x_D^2*x_E*y_C^4*y_D+8*x_C*x_D^2*x_E*y_C^2*y_D^3+4*x_C*x_D^2*x_E*y_C^2*y_D-12*x_C*x_D^2*x_E*y_C*y_D^4-4*x_C*x_D^2*x_E*y_C*y_D^2-x_C*x_D^2*y_C^4*y_D^3+3*x_C*x_D^2*y_C^3*y_D^4-3*x_C*x_D^2*y_C^3*y_D^2-2*x_C*x_D^2*y_C^2*y_D^5+3*x_C*x_D^2*y_C^2*y_D^3+x_C*x_D*x_E^2*y_C^4*y_D+2*x_C*x_D*x_E^2*y_C^3*y_D^2-6*x_C*x_D*x_E^2*y_C^2*y_D^3-3*x_C*x_D*x_E^2*$

$y_C^2*y_D+9*x_C*x_D*x_E^2*y_C*y_D^4+5*x_C*x_D*x_E^2*y_C*y_D^2+4*x_C*x_D*x_E*y_C^4*y_D^3-12*x_C*x_D*x_E*y_C^3*y_D^4+4*x_C*x_D*x_E*y_C^3*y_D^2+8*x_C*x_D*x_E*y_C^2*y_D^5-4*x_C*x_D*x_E*y_C^2*y_D^3-3*x_C*x_E^2*y_C^4*y_D^3+9*x_C*x_E^2*y_C^3*y_D^4-x_C*x_E^2*y_C^3*y_D^2-6*x_C*x_E^2*y_C^2*y_D^5+x_C*x_E^2*y_C^2*y_D^3-x_D^7*y_C^3+4*x_D^6*x_E*y_C^3+x_D^6*y_C^3+x_D^6*y_C^2*y_D-3*x_D^5*x_E^2*y_C^3-8*x_D^5*x_E*y_C^3+4*x_D^5*x_E*y_C^2*y_D+3*x_D^5*y_C^4*y_D-2*x_D^5*y_C^3*y_D^2+x_D^5*y_C^3-3*x_D^5*y_C^2*y_D+7*x_D^4*x_E^2*y_C^3-5*x_D^4*x_E^2*y_C^2*y_D-4*x_D^4*x_E*y_C^4*y_D+4*x_D^4*x_E*y_C^3-4*x_D^4*x_E*y_C^2*y_D-3*x_D^4*y_C^4*y_D-x_D^4*y_C^3+2*x_D^4*y_C^2*y_D^3+2*x_D^4*y_C^2*y_D+x_D^3*x_E^2*y_C^4*y_D+2*x_D^3*x_E^2*y_C^3*y_D^2-5*x_D^3*x_E^2*y_C^3+7*x_D^3*x_E^2*y_C^2*y_D+4*x_D^3*x_E*y_C^4*y_D-2*x_D^3*y_C^5*y_D^2+3*x_D^3*y_C^4*y_D^3-x_D^3*y_C^3*y_D^4+3*x_D^3*y_C^3*y_D^2-3*x_D^3*y_C^2*y_D^3-x_D^2*x_E^2*y_C^4*y_D+x_D^2*x_E^2*y_C^3-2*x_D^2*x_E^2*y_C^2*y_D^3-2*x_D^2*x_E^2*y_C^2*y_D+4*x_D^2*x_E*y_C^4*y_D^3-4*x_D^2*x_E*y_C^3*y_D^4-4*x_D^2*x_E*y_C^3*y_D^2+4*x_D^2*x_E*y_C^2*y_D^3+x_D^2*y_C^5*y_D^2-x_D^2*y_C^4*y_D^3-x_D^2*y_C^3*y_D^4+x_D^2*y_C^2*y_D^5+2*x_D*x_E^2*y_C^5*y_D^2-7*x_D*x_E^2*y_C^4*y_D^3+5*x_D*x_E^2*y_C^3*y_D^4+x_D*x_E^2*y_C^3*y_D^2-x_D*x_E^2*y_C^2*y_D^3-4*x_D*x_E*y_C^4*y_D^3+8*x_D*x_E*y_C^3*y_D^4-4*x_D*x_E*y_C^2*y_D^5-x_E^2*y_C^5*y_D^2+5*x_E^2*y_C^4*y_D^3-7*x_E^2*y_C^3*y_D^4+3*x_E^2*y_C^2*y_D^5)/(y_C*y_D*(2*x_D-1)*(x_D+x_E-2*x_C*x_D+2*x_C*x_E-4*x_D*x_E)*(x_D*y_C-x_D*y_D+x_E*y_C-x_E*y_D-x_D^3*y_C+2*x_D^2*y_D+2*x_C*x_D^2*y_C-2*x_C*x_D^2*y_D+x_C^2*x_D*y_D+x_C^2*x_E*y_D+3*x_D^2*x_E*y_C-3*x_D*y_C*y_D^2+3*x_D*y_C^2*y_D+x_E*y_C*y_D^2-x_E*y_C^2*y_D-2*x_C*x_D*y_C+2*x_C*x_E*y_C-4*x_D*x_E*y_C+2*x_D*x_E*y_D-2*x_C*x_D*x_E*y_C-2*x_C*x_D*x_E*y_D))$.
If we substitute the values of x_D and y_D into these one can easily verify that each of the expressions evaluates to 0 as desired, which proves indeed that lines NSV, OTW, PUZ, and QR are concurrent.

□

Schiffler's theorem follows as the special case of this by taking $E = A$.

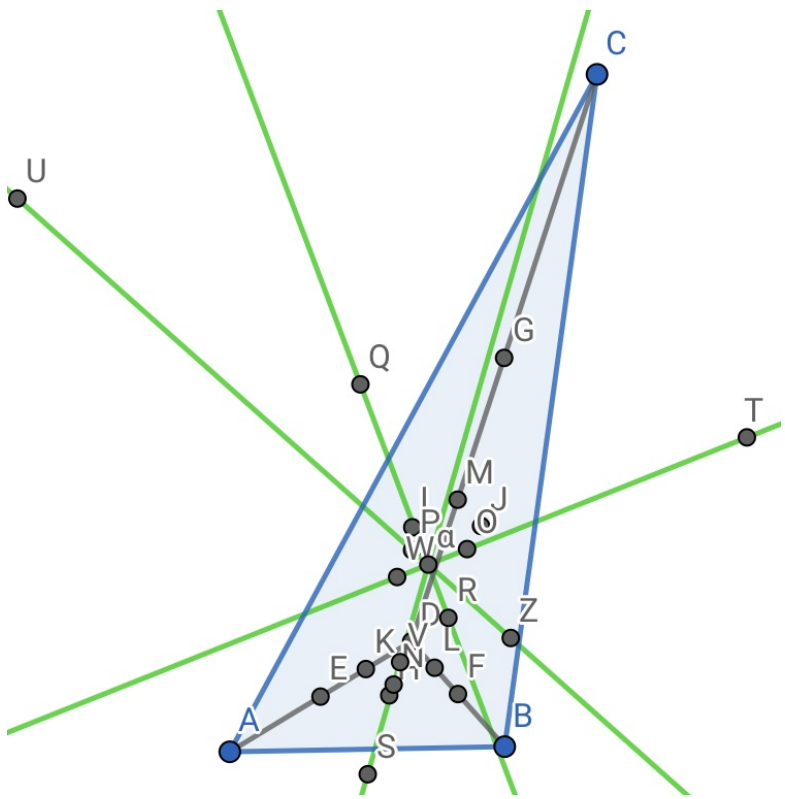

FIGURE 1. Illustration of theorem 2.1 where we take E to be the midpoint of A and D.

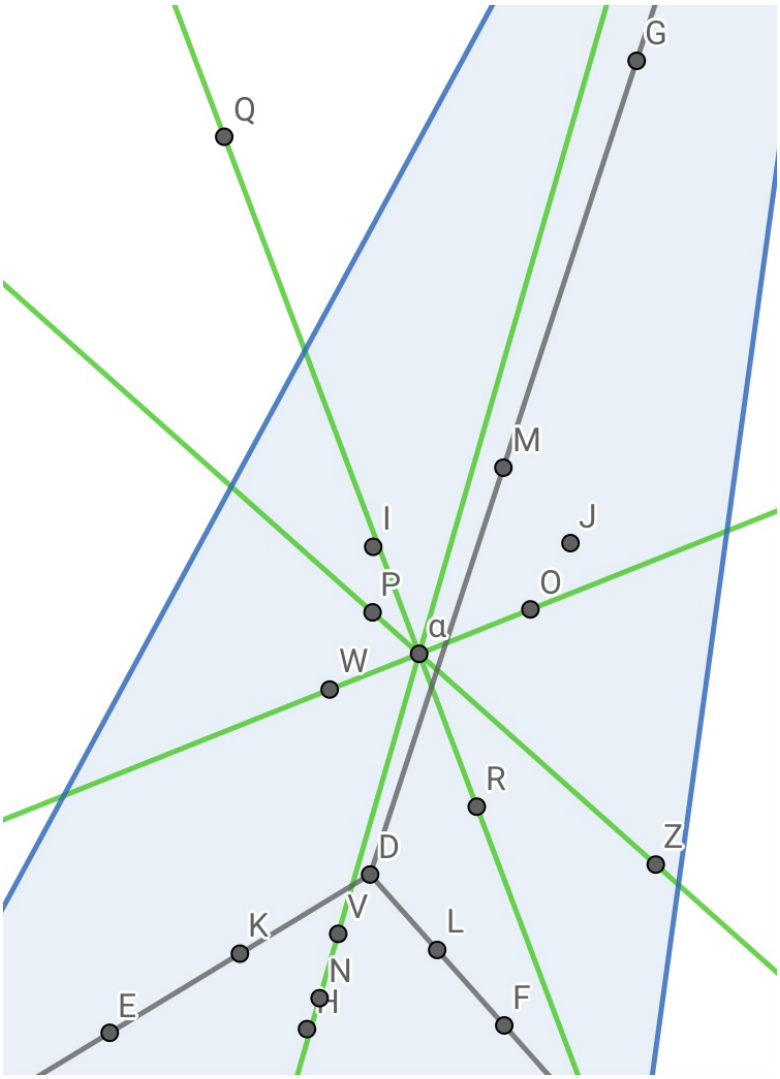

Figure 2. Zoomed in illustration of Figure 1.

References

[1] Weisstein, Eric W. "Schiffler Point." From MathWorld–A Wolfram Web Resource. https://mathworld.wolfram.com/SchifflerPoint.html

[2] Warren, B.L., "Generalization of the Triangle Centroid and the Van Lamoen Circle", *International Journal of Computer Discovered Mathematics*, Vol. 9 (2024), pp. 44-53.

[3] Warren, B.L., "Generalization of the Euler and Nagel Lines", *International Journal of Computer Discovered Mathematics*, Vol. 9 (2024), pp. 165-171.