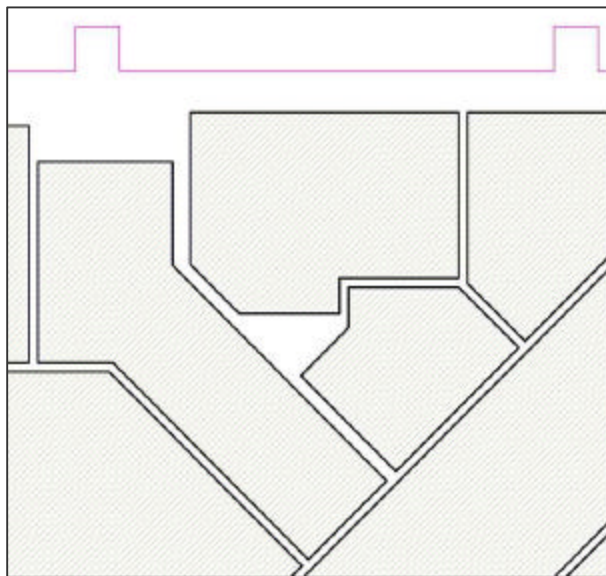


Die Wände in Visio haben mehrere Funktionen, einmal sind sie „Träger“ der sie beinhaltenden Türen und Fenster, die in der Regel automatisch angepasst werden, sie verschneiden sich an ihren Enden so, dass keine Lücken entstehen und, eine ganz wesentliche Funktion, wenn man ein Raum-Shape innerhalb eines Wandpolygons zeichnet, wird dieser Raum automatisch an die umfassenden Wände angepasst. Ein wesentlicher Nachteil dieses Verfahrens ist, dass im Hintergrund ein so großer Rechenaufwand betrieben muss, dass dieser Mechanismus für große, reale Gebäude nicht machbar ist. Das ist einer der Gründe, weshalb praktisch alle professionellen Gebäudeinformationssysteme ausschließlich mit Räumen arbeiten.

Selbst wenn man mit den Wand-Shapes von Visio arbeiten wollte, hätte man Probleme, wenn

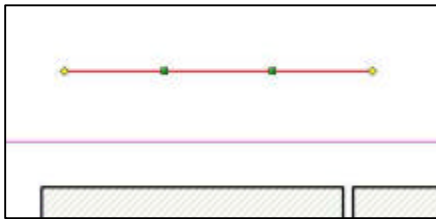


man die in der Abbildung dargestellte Situation vorfinden würde. Will man jetzt die Flächen außerhalb der Räume, die nach der DIN 277 in der Regel als Konstruktions-Grundflächen anzusehen sind, und auch im Abzugsverfahren als Differenz der Brutto-Grundfläche und der Netto-Grundfläche (also der Gesamtumriss des Gebäudes minus der Fläche der Räume) bietet sich als Alternative an, den freien Raum zwischen den Raum-Shapes ebenfalls als flächenbehaftete Shapes zu

erfassen und beispielsweise einem Layer „Wand“ a. ä. zuzuweisen. Die Fläche dieser „Ersatzwände“ werden farblich gekennzeichnet und können später getrennt auch als Konstruktionsgrundfläche berechnet werden.

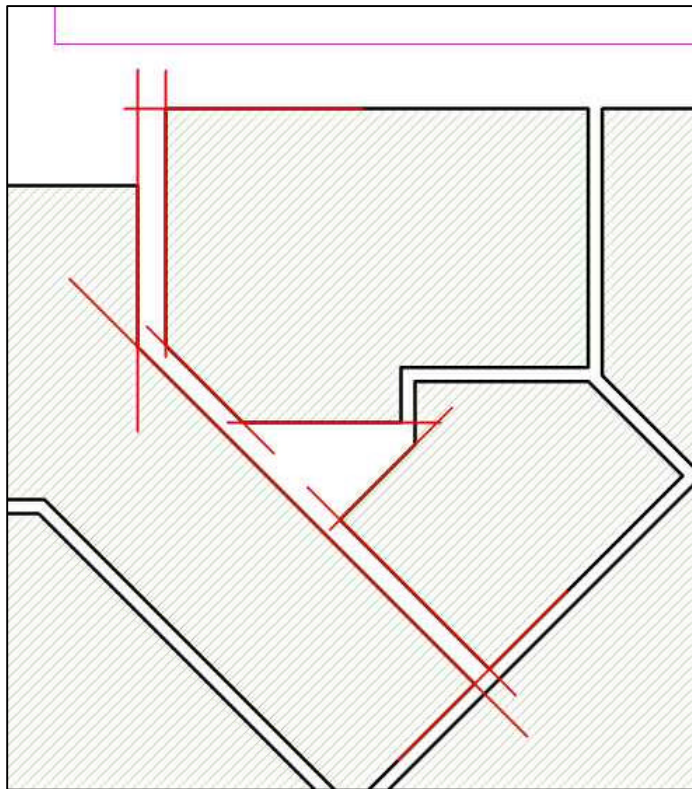
Ein elegantes und gleichzeitig genaues Verfahren soll hier dargestellt werden. Man könnte die Zwischenräume z. B. mit dem Bleistift-Symbol von Visio zeichnen, man könnte die Zwischenräume mit Begrenzungslinien versehen und deren Schnittpunkte zu einer Polylinie

verbinden, wobei man das Problem hat, die Linien genau zu treffen. Auch hier ist die Generierung der Polylinie etwas mühsam.



Wir modifizieren das letzte Verfahren. Visio verfügt in der Schablone „Zeichentools“ über eine sog. „Linie mit Verlängerung“, die den Vorteil hat, wegen der Einrastpunkte sehr gut die umfahrenden Linien der Räume einfangen zu können. Das wird als nächstes

gemacht.



Als Ergebnis erhält man die nebenstehende Grafik. Sie ist der Ausgangspunkt und Kernpunkt des hier vorgestellten Tools. Der Bereich bzw. die Fläche, die die Wand darstellen soll, ist jetzt mit den (rot) gekennzeichneten Linien umgeben.

Als nächstes werden nun die **Schnittpunkte** der Geraden miteinander ermittelt. Danach wird die Visio-Funktion **DrawPolyline** aufgerufen, die aus den übergebenen X- und Y-Koordinaten der Schnittpunkte ein räumliches Shape

erzeugt, welches mit einer eigenen Farbe belegt und auf einen separaten Layer gelegt wird.

Erläuterung zu dem Verfahren:

Mathematisch ist eine Gerade eindeutig bestimmt, entweder durch zwei Punkte, oder durch einen Punkt und ihre Steigung. Für alle erzeugten Hilfslinien sind sowohl zwei Punkte als

auch deren Steigung bekannt. Das kann man leicht nachprüfen, indem man eine der

1-D Endpoints			
BeginX	23.6279 m	EndX	23.9862 m
BeginY	43.9945 m	EndY	43.6362 m
Shape Transform			
Width	0.5068 m		Pin
Height	0.0000 m		Pin
Angle	-44.9999 deg		LocPin

Hilfslinien
markiert und
das sog.
ShapeSheet
einblendet.

Für uns sind die beiden Anfangs- und Endpunkte für die X-Achse – *BeginX*, *EndX*, bzw. *BeginY* und *EndY* von Interesse. Mit Hilfe dieser Punkte werden die Geradengleichungen und anschließend deren Schnittpunkte ermittelt.

Die Zweipunkteform der Geradengleichung lautet:

$$\frac{Y_2 - Y_1}{Y - Y_1} = \frac{X_2 - X_1}{X - X_1}$$

Umgeformt ergibt sich:

$$Y = \frac{(Y_2 - Y_1)}{(X_2 - X_1)} \cdot (X - X_1)$$

oder:

$$Y = m x + b$$

wobei

$$m = \frac{(Y_2 - Y_1)}{(X_2 - X_1)}$$

m die Steigung der Geraden angibt

und

$$b = Y_1 - m x_1$$

b den Schnittpunkt der Geraden mit der Y-Achse.

Da beide Geraden einen gemeinsamen Punkt, den Schnittpunkt, haben, kann man ihre Gleichungen

$$Y = m_1 X - b_1$$

$$Y = m_2 X - b_2$$

gleichsetzen

$$m_1 X_i - b_1 = m_2 X_i - b_2$$

Durch Umformung erhält man die gesuchten X- und Y-Koordinaten des jeweiligen

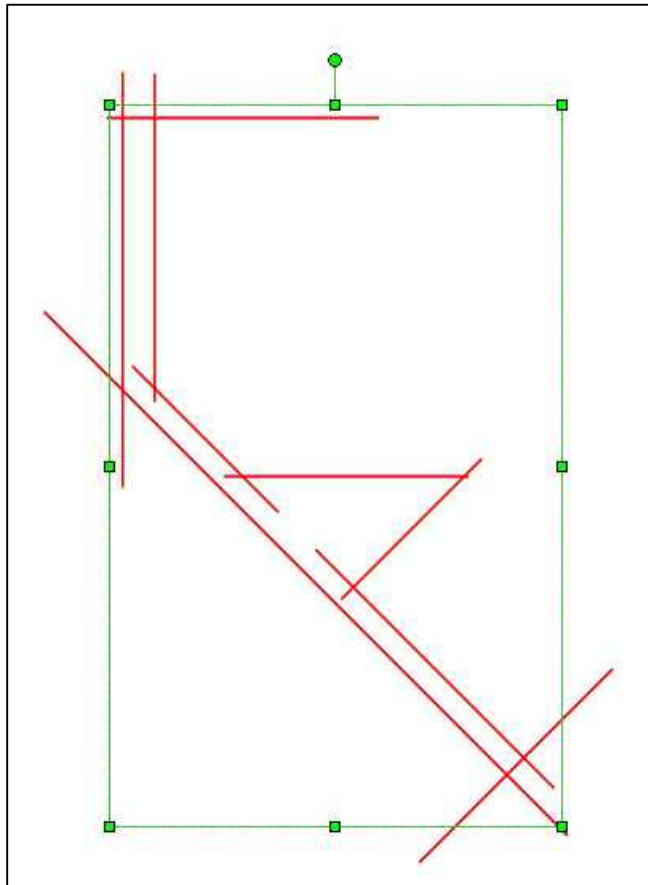
Schnittpunktes

$$X_i = \frac{b_2 - b_1}{m_1 - m_2}$$

$$Y_i = \frac{b_2 m_1 - b_1 m_2}{m_1 - m_2}$$

Die im Folgenden wiedergegebene Prozedur, die unter einem beliebigen Namen in einer Schablone angespeichert wird, funktioniert nun wie folgt:

- Nach Erzeugen der Hilfslinien müssen diese im Uhrzeiger- oder im Gegenuhrzeigersinn – wichtig ist alleine, dass alle Linien nacheinander angewählt



werden – markiert werden. Am besten ist, man schaltet alle anderen Layer aus.

- Jetzt wird das Makro aufgerufen, welches folgende Schritte ausführt
- Für jede der angewählten Linien wird, wenn eine benutzerdefinierte Abteilung existiert, diese gelöscht und neu erzeugt
- Jede der Abteilungen erhält sechs Zeilen. In diese werden oben erläuterte Rechenschritte als Formeln eingetragen. Dabei wird

der Sonderfall, dass die Steigung Null ist ($\text{Anfang}_X = \text{Ende}_X$), berücksichtigt

- Es wird ein zweidimensionales Feld mit den X- und Y-Koordinaten der Schrittpunkte erzeugt
- Anschließend wird der Layer „Wand_neu“, falls vorhanden, gelöscht und neu erstellt. Er wird auf aktiv gestellt, d. h. alle jetzt erzeugten Shapes werden auf diesen Layer gelegt.

- Die Visio-Funktion **Drawpolyline** wird aufgerufen. Als Argument wird ihr das Feld-Array mitgegeben.

```
Public Sub Mach_Säule_Visio_2007()

Dim sh As Visio.Shape
Dim c As Visio.Cell
Dim pg As Visio.Page
Dim sel As Visio.Selection

Dim i As Integer
Dim j As Integer
Dim z As Integer
Dim Anzahl As Integer
Dim Layer_besteht As String
Dim Feld() As Double
Dim b1 As Double
Dim b2 As Double
Dim x1 As Double
Dim y1 As Double
Dim m1 As Double
Dim m2 As Double
Dim Teiler As Double

.....

Dim Anfang_X As Double, Ende_X As Double
Dim Anfang_Y As Double, Ende_Y As Double
.....

Dim Ecken_X() As Double
Dim Ecken_Y() As Double

Set sel = ActiveWindow.Selection
Anzahl = sel.Count
For i = 1 To Anzahl
    Set sh = sel(i)
    If sh.SectionExists(243, 1) Then
        sh.DeleteSection (243)
    End If
Next i

For i = 1 To Anzahl
    Set sh = sel(i)
```

```
sh.AddSection (243)
Next i

For i = 1 To Anzahl
  Set sh = sel(i)
  For j = 1 To 6
    sh.AddRow 243, visSectionProp + j - 1, 0
  Next j
Next i

For i = 1 To Anzahl
  Set sh = sel(i)

  .....

  Set c = sh.Cells("AnfangX")
  Anfang_X = c.ResultIU
  Set c = sh.Cells("EndeX")
  Ende_X = c.ResultIU
  If Anfang_X = Ende_X Then
    Anfang_X = Anfang_X + (Anfang_X / 100000)
    Set c = sh.Cells("AnfangX")
    c.ResultIU = Anfang_X
  End If

  .....

  Set c = sh.Cells("Prop.Row_1.Value")
  c.Formula = "EndeY-AnfangY"

  Set c = sh.Cells("Prop.Row_2.Value")
  c.Formula = "EndeX-AnfangX"

  Set c = sh.Cells("Prop.Row_3.Value")
  c.Formula = "Prop.Row_1/Prop.Row_2"

  Set c = sh.Cells("Prop.Row_4.Value")
  c.Formula = "ARCTAN(Prop.Row_3)"

  Set c = sh.Cells("Prop.Row_5.Value")
  c.Formula = "TAN(Prop.Row_4)"

  Set c = sh.Cells("Prop.Row_6.Value")
  c.Formula = "AnfangY-Prop.Row_5*AnfangX"
```

```
Next i

ReDim Feld(1 To (Anzahl + 1) * 2) As Double
'Ecken = (Anzahl + 1) * 2

ReDim Ecken_X(Anzahl + 1) As Double
ReDim Ecken_Y(Anzahl + 1) As Double

For i = 1 To Anzahl
    Set sh = sel(i)
    Set c = sh.Cells("Prop.Row_5.Value")
    m1 = c.ResultIU

    Set c = sh.Cells("Prop.Row_6.Value")
    b1 = c.ResultIU
    b1 = b1 / 39.3701

    If i = Anzahl Then
        Set sh = sel(1)
    End If
    If i < Anzahl Then
        Set sh = sel(i + 1)
    End If
    Set c = sh.Cells("Prop.Row_5.Value")
    m2 = c.ResultIU

    Set c = sh.Cells("Prop.Row_6.Value")
    b2 = c.ResultIU
    b2 = b2 / 39.3701

    Teiler = m1 - m2

    .....
    x1 = (b2 - b1) / Teiler
    y1 = (b2 * m1 - b1 * m2) / Teiler
    Debug.Print "x" & i & " ="; x1
    Debug.Print "y" & i & " ="; y1
    Ecken_X(i) = x1
    Ecken_Y(i) = y1
    .....
'Stop

Next i
```



```
Ecken_X(Anzahl + 1) = Ecken_X(1)
Ecken_Y(Anzahl + 1) = Ecken_Y(1)

For z = 1 To (Anzahl + 1) * 2
  If z Mod 2 = 1 Then
    i = z / 2 + 0.5
    Feld(z) = Ecken_X(i) * 39.3701
  End If
  If z Mod 2 = 0 Then
    j = z / 2
    Feld(z) = Ecken_Y(j) * 39.3701
  End If
Next z

Layer_besteht = "N"
Set pg = ActivePage
For i = 1 To pg.Layers().Count
  If pg.Layers(i).Name = "Wand_neu" Then
    Layer_besteht = "J"
    Set c = pg.Layers(i).CellsC(6)
    c.Formula = "1"
  End If
Next i

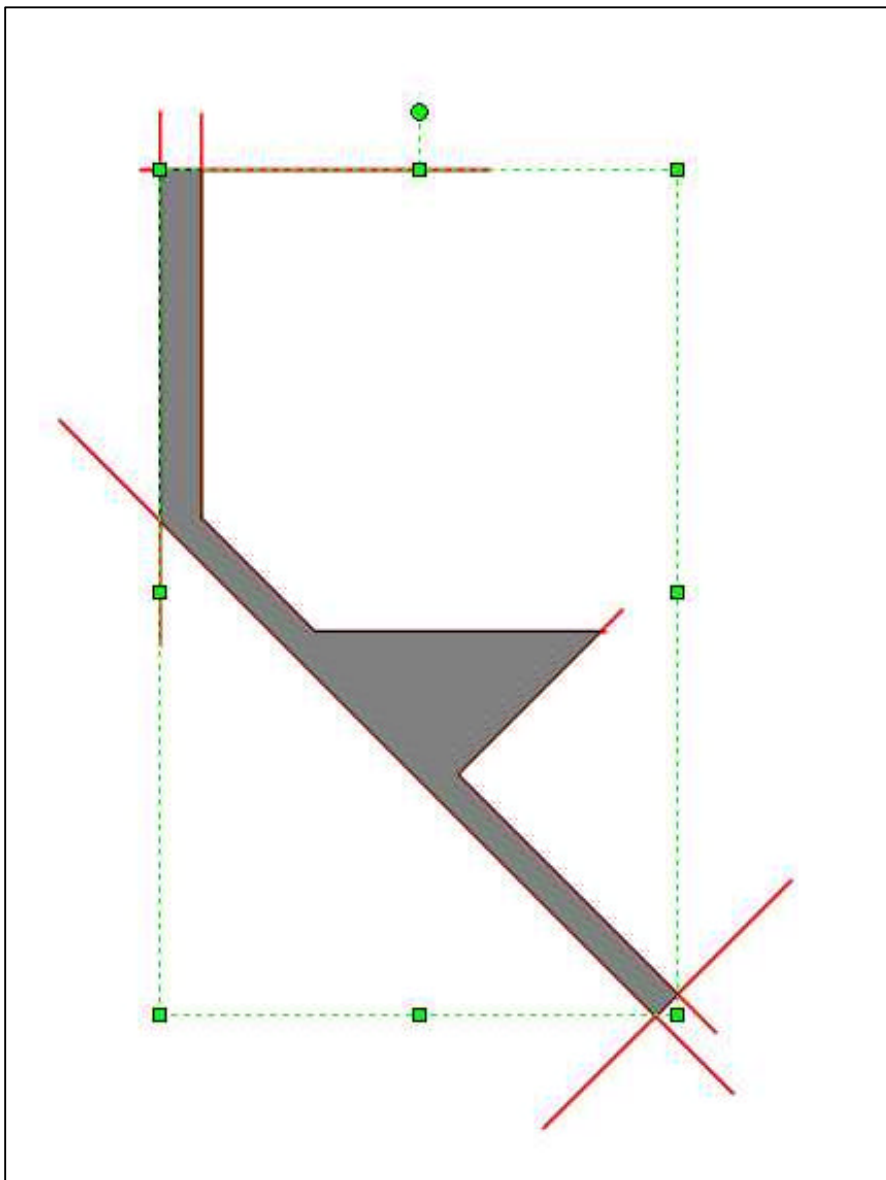
If Layer_besteht = "N" Then
  pg.Layers.Add ("Wand_neu")
  Set c = pg.Layers("Wand_neu").CellsC(6)
  c.Formula = "1"
End If

Set sh = pg.DrawPolyline(Feld, 0)
Set c = sh.Cells("LinienBreite")
c.Formula = "0,04 pt"
'c.Formula = "0,04 pkt"
Set c = sh.Cells("FüllVGrund")
c.Formula = "19"

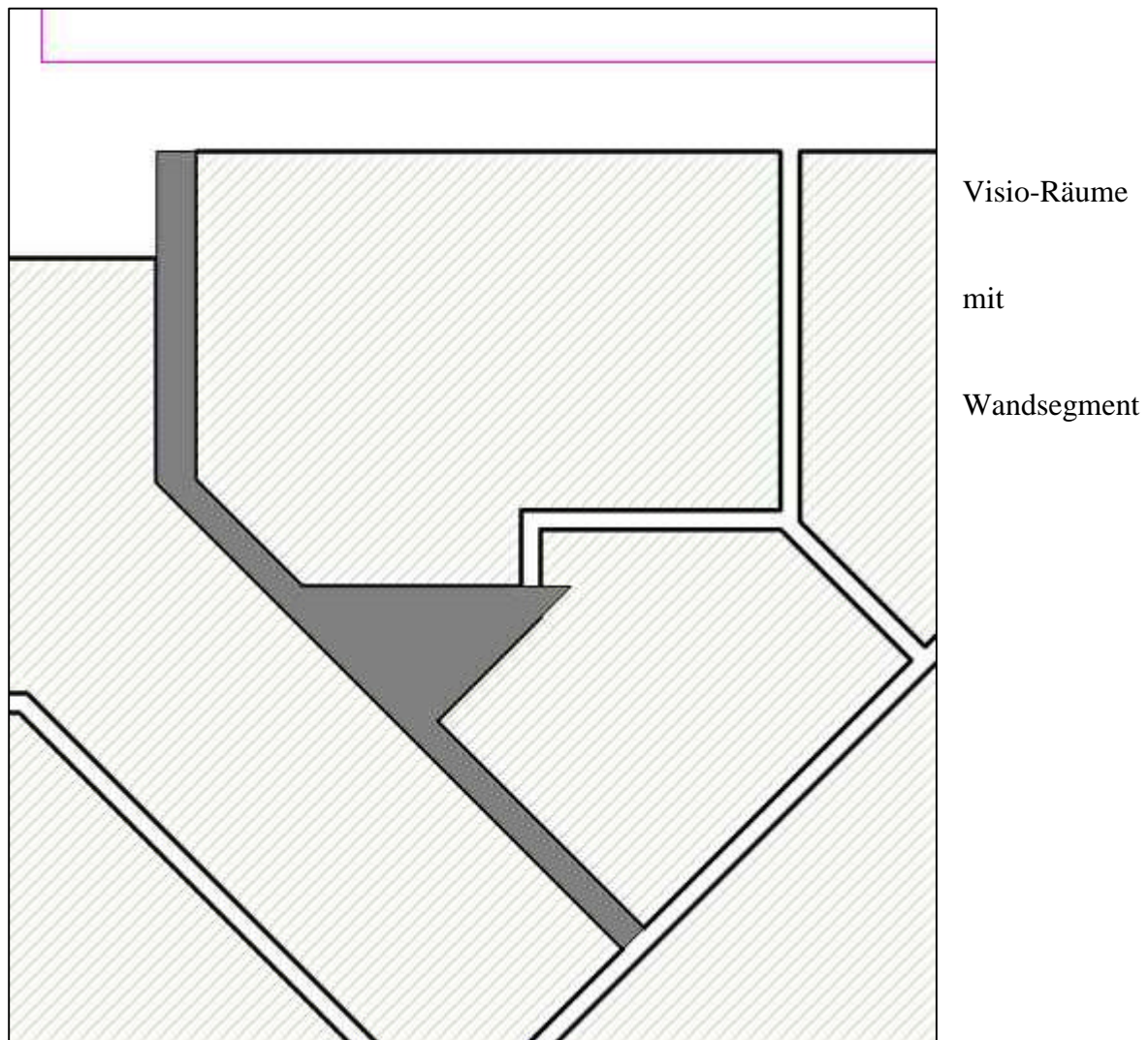
Set c = pg.Layers("Wand_neu").CellsC(6)
c.Formula = "0"
```

End Sub

Als Ergebnis der Operation erhalten Sie die folgende Darstellung



Wenn Sie die Hilfslinien wieder aus- und die restlichen Layer wieder einschalten, erhalten Sie Raum-Shapes mit einem passgenau erzeugtem Shape, welches in unserem Fall ein Wandsegment repräsentieren soll.



Wenn Ihnen der Aufwand zu groß erscheinen sollte, ein einfaches Flächenmodul zu erstellen, denken Sie an die komfortable Bedienung und daran, dass diese Prozedur ein universelles Werkzeug zum Digitalisieren von komplexen geometrischen Objekten darstellt.