the PERSON

Wanda Rutkiewicz - 4 February 1943 – 12–13 May 1992) was a Polish mountaineer and computer engineer. She was the first woman to reach the summit of K2 and the third woman (first European woman) to summit Mount Everest.

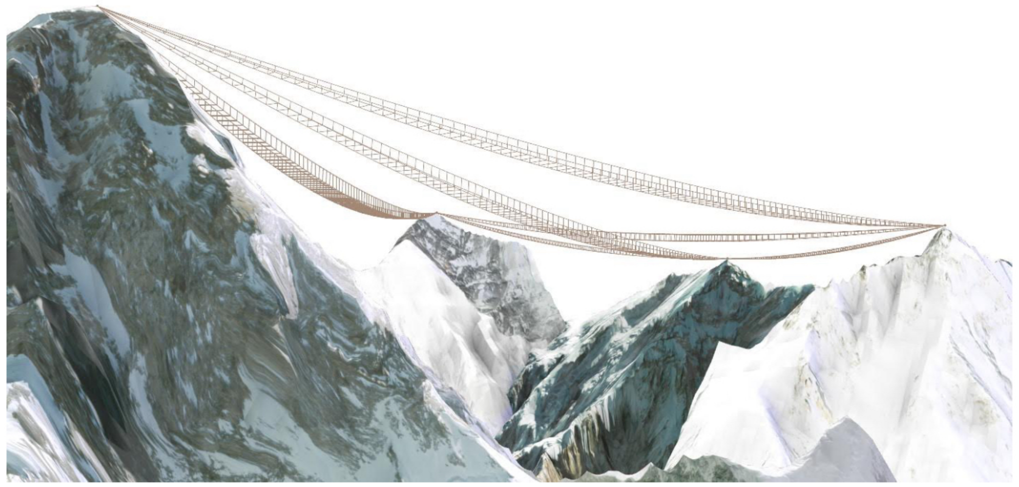
She was one of history's best female alpinist. I chose four of the most important mountains she summited. The ones over 8000m are Mount Everest, Nanga Parbat, K2, Shisha Pangma, Gasherbrum I, Gasherbrum II, Cho Oju and Annapurna.

© Archiv Wanda Rutkiewicz

the PROJECT

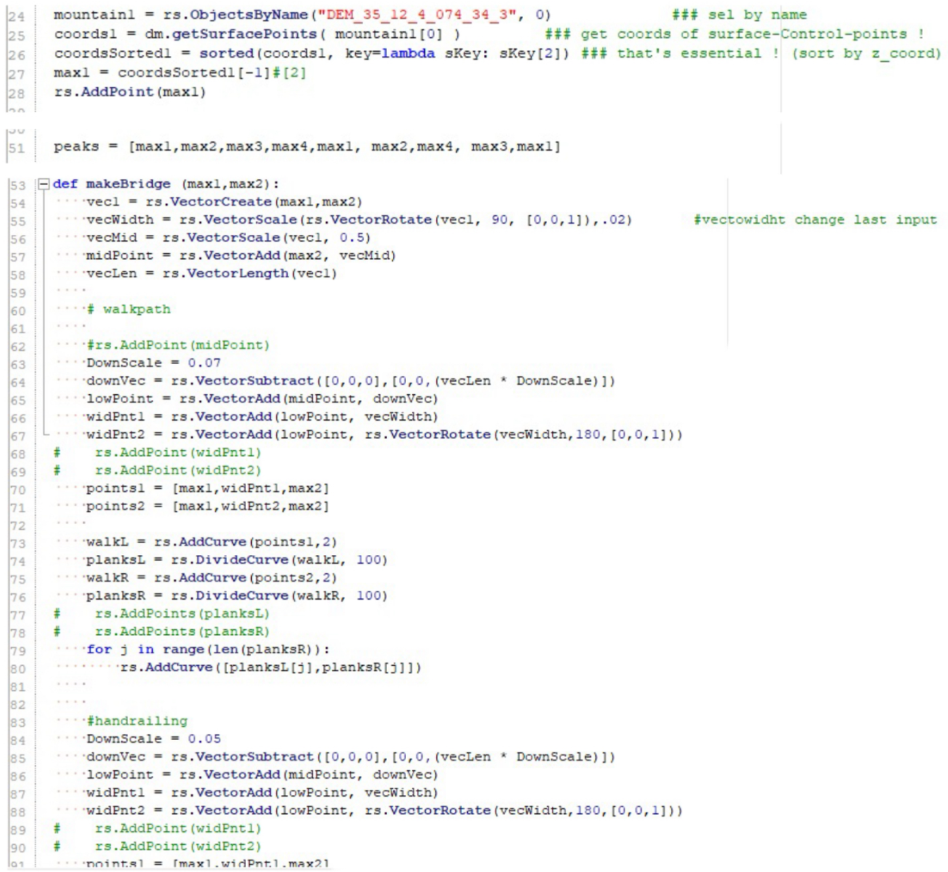
In the memory of her achievments I wanted to make a collage of surfaces of some of the highest mountains in the world (Annapurna, Nanga Parbat, K2 and Mount Everest), she summited with suspension brigdes.

The bridges connect the highest points of the mountain models. Their width and suspension length depends on the distances between the mountains. The models are in tehir original geographical position.



the CODE

First step of the code is to define the highest points of four mountains that are in a seperate rhino file, imported in their geographical position. The rest is based on a definition to make a bridge, which is repeatedly used to create bridges with hadrailings and planks between defined peaks. The suspension of the brigde depends on brigde's length that's why there's no collision of two brigdes crossing each other paths.



```
mountain1 = rs.ObjectsByName("DEM_35_12_4_074_34_3", 0)            ### sel by name
coords1 = dm.getSurfacePoints( mountain1[0] )          ### get coords of surface-Control-points !
coordsSorted1 = sorted(coords1, key=lambda sKey: sKey[2]) ### that's essential ! (sort by z_coord)
max1 = coordsSorted1[-1]#[2]
rs.AddPoint(max1)

peaks = [max1,max2,max3,max4,max1, max2,max4, max3,max1]

def makeBridge (max1,max2):
    vec1 = rs.VectorCreate(max1,max2)
    vecWidth = rs.VectorScale(rs.VectorRotate(vec1, 90, [0,0,1]),.02)       #vectowidht change last input
    vecMid = rs.VectorScale(vec1, 0.5)
    midPoint = rs.VectorAdd(max2, vecMid)
    vecLen = rs.VectorLength(vec1)
    
    # walkpath
    
    #rs.AddPoint(midPoint)
    DownScale = 0.07
    downVec = rs.VectorSubtract([0,0,0],[0,0,(vecLen * DownScale)])
    lowPoint = rs.VectorAdd(midPoint, downVec)
    widPnt1 = rs.VectorAdd(lowPoint, vecWidth)
    widPnt2 = rs.VectorAdd(lowPoint, rs.VectorRotate(vecWidth,180,[0,0,1]))
#    rs.AddPoint(widPnt1)
#    rs.AddPoint(widPnt2)
    points1 = [max1,widPnt1,max2]
    points2 = [max1,widPnt2,max2]
    
    walkL = rs.AddCurve(points1,2)
    planksL = rs.DivideCurve(walkL, 100)
    walkR = rs.AddCurve(points2,2)
    planksR = rs.DivideCurve(walkR, 100)
#    rs.AddPoints(planksL)
#    rs.AddPoints(planksR)
    for j in range(len(planksR)):
        rs.AddCurve([planksL[j],planksR[j]])
    
    
    #handrailing
    DownScale = 0.05
    downVec = rs.VectorSubtract([0,0,0],[0,0,(vecLen * DownScale)])
    lowPoint = rs.VectorAdd(midPoint, downVec)
    widPnt1 = rs.VectorAdd(lowPoint, vecWidth)
    widPnt2 = rs.VectorAdd(lowPoint, rs.VectorRotate(vecWidth,180,[0,0,1]))
#    rs.AddPoint(widPnt1)
#    rs.AddPoint(widPnt2)
    points1 = [max1,widPnt1,max2]
```

Weronika Skorupa
DM2 | Finale

Gruppe 2 | Gruber
WS | 2024/25