R1 = 0.6

R2 = 100

R3 = 0.35

PR = 0.8

nayborsafe3D<- function(v, i, j, k, M, N, L){

n = 1

ind = i + (M\*((j-1) + N\*(k-1)))

inds = ind

if (i > 1){ inds = c(inds,ind-1) }

if (i < M){ inds = c(inds,ind+1) }

if (j > 1){ inds = c(inds,ind-M) }

if (j < N){ inds = c(inds,ind+M) }

if (k > 1){ inds = c(inds,ind-(M\*N)) }

if (k < L){ inds = c(inds,ind+(M\*N)) }

ninds = which(v[inds] == -1)

if (length(ninds) > 0) n = 0

return(n)

}

# function for3 measurements, M1, M2 and M3, between min and max

violationMap <- function(minM1, maxM1, M, minM2, maxM2, N, minM3, maxM3, L, R1, R2, R3){

mat <- matrix(1, ncol = 4, nrow = M\*N\*L)

numsafe = 0

R1fail = 0

R2fail = 0

R3fail = 0

m1step = (maxM1 - minM1)/M

m2step = (maxM2 - minM2)/N

m3step = (maxM3 - minM3)/L

print(c("msteps:", m1step, m2step, m3step))

for (i in 1:M){

for (j in 1:N){

for (k in 1:L){

M1 = minM1 + m1step\*i

M2 = minM2 + m2step\*j

M3 = minM3 + m3step\*k

ind = i + (M\*((j-1) + N\*(k-1)))

mat[ind,1] = M1

mat[ind,2] = M2

mat[ind,3] = M3

P1 = 0.2018 + (0.8191 \* M1)

P2 = 0.414 + (0.4612 \* M1)

P3 = 0.1

P4 = -0.1618+ (1.2523 \* M3)

T1 = 0

T2 = 19.765 + (15.627 \* M1)

T3 = 63.15

T1F = 0

T2F = 7.343 + ( 72.234 \* M1) + (161.485 \* M2) - (231.337 \* M1\*M2)

TR = 6.498 - (5.482 \* M1) - (6.855 \* M3) + (8.301 \* M1\*M3)

# start safe

mat[ind,4] = 1

r1 = (P1\*P2\*P4) / (1 - PR\*P3)

r2 = (P1\*(P3\*T2 - P2\*T3 - P3\*T2F - T1 - T2 + T1F) + PR\*P3\*(T1F - P1\*TR - P1\*T1F) - T1F) / (PR\*P3 - 1)

r3 = (P1\*P2\*(1 - P4)) / (1 - PR\*P3)

if ((1 - PR\*P3) == 0) {

mat[ind,4] = -1

}

else {

if (r1 < R1){

mat[ind,4] = -1

R1fail = R1fail+1

}

if (r2 > R2) {

mat[ind,4] = -1

R2fail = R2fail+1

}

if (r3 > R3) {

mat[ind,4] = -1

R3fail = R3fail+1

}

if (mat[ind,4] != -1){

numsafe = numsafe +1

}

}

}

}

}

numviolations = (M\*N\*L) - numsafe

# now find border pixels

numborder = 0

numedge = 0

for (i in 1:M){

for (j in 1:N){

for (k in 1:L){

ind = i + (M\*((j-1) + N\*(k-1)))

if (mat[ind,4] == 1){

if (nayborsafe3D(mat[,4], i, j, k, M, N, L) == 0) {

mat[ind,4] = 0

numborder = numborder + 1

}

else {

if ((i==1)| (j ==1)|(k ==1)|(i==M)|(j == N)|(k == L)){

mat[ind,4] = 2

numedge = numedge + 1

}

}

}

}

}

}

print(c("safe: ", numsafe, "border: ", numborder, "edge: ", numedge, "violations: ", numviolations, "R1fail: ", R1fail, "R2fail: ", R2fail, "R3fail: ", R3fail))

return(mat)

}

vmap = violationMap(0.5, 1.0, 40, 0.25, 0.5, 40, 0.5, 1.0, 40, R1, R2, R3)

colnames(vmap) = c("M1", "M2", "M3", "SBV")

write.csv(vmap, "violationMap.csv", row.names = F)

#PLOTTING:

install.packages("rgl")

no

library("rgl")

# safe, border and violation colours

c = vMap[,4]+2

open3d()

points3d(vMap[,1], vMap[,2], vMap[,3], color = c)

# just safe vs violation

c2 = c

c2[which(c2 == 3)] = 2

open3d()

points3d(vMap[,1], vMap[,2], vMap[,3], color = c2)