Vignette for Bayesian Spatiotemporal Wombling

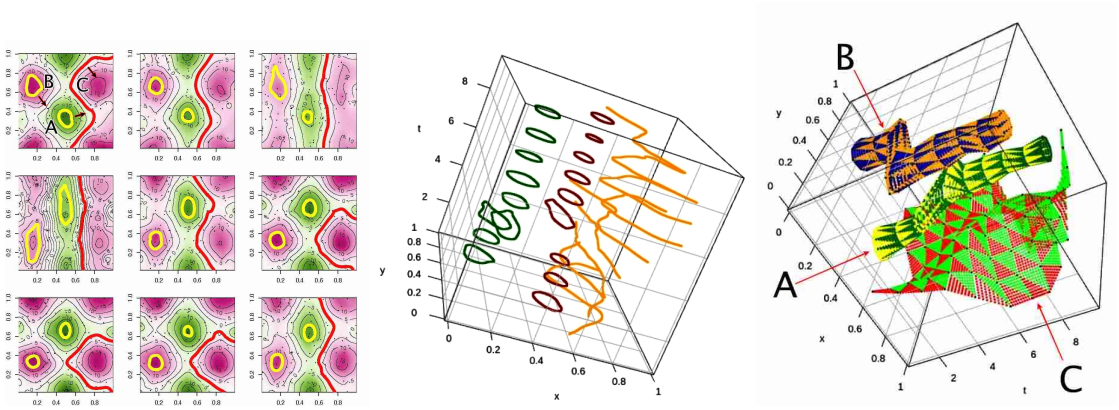
Simulated Experimental Data

1 Selecting Curves

The following code will generate Figure 1 (a). We used the `spWombling` R-package to lift contours. Annotated curves can be picked using `locator(100)` on each spatiotemporal surface plots.

```
require(sptwombling)
set.seed(1234)
Ns = 100; Nt = 9; N = Ns*Nt
t = seq(1,Nt, length.out=Nt)
coords = matrix(runif(2*Ns),nc=2)
tau = 1

# plot(coords, xlab="coords.x",ylab="coords.y")
# co-ordinates are observed for each time point
sim.pattern = array(NA, c(Ns,Nt))
## create synthetic y
for(j in 1:9){
  sim.pattern[,j] = rnorm(Ns,
                          mean = 10 * (sin(3 * pi * coords[,1])
                                + cos(3 * pi * coords[,2])
                                * cos(t[j] * pi/7)),
                          sd = tau)
}
y = c()
for(i in 1:100){
  y = c(y,sim.pattern[i,1:9])
}
mean(y); var(y); range(y)

require(raster)
require(sp)
require(spWombling)
wb.1 = wb.2 = wb.0 = list()
id.lines = id.levels = matrix(NA, nrow = 9, ncol = 2)
```

```
length.p = 10
tr.points = 10 # change to 20 to generate plots
############################
# Selecting Closed Curves #
############################
id.lines[1,] = c(1, 2)
id.lines[2,] = c(2, 2)
id.lines[3,] = c(1, 1)
id.lines[4,] = c(1, 1)
id.lines[5,] = c(1, 2)
id.lines[6,] = c(1, 2)
id.lines[7,] = c(1, 2)
id.lines[8,] = c(1, 2)
id.lines[9,] = c(1, 2)

id.levels[1, ] = c("-16", "16")
id.levels[2, ] = c("-14", "14")
id.levels[3, ] = c("-12", "10")
id.levels[4, ] = c("-10", "10")
id.levels[5, ] = c("-14", "14")
id.levels[6, ] = c("-16", "16")
id.levels[7, ] = c("-18", "18")
id.levels[8, ] = c("-18", "16")
id.levels[9, ] = c("-14", "14")

########################
# Selecting Open Curve #
########################
id.lines.0 = rep(4, 9)
id.lines.0[4] = 3

par(mfrow = c(3, 3))
for(j in 1:Nt){
  par(mar=rep(2,4))
  raster.obj =  sp_plot(11,"PiYG",
                        cbind(coords,sim.pattern[,j]),
                        zlim = c(min(y) - 0.5, max(y) + 0.5),
                        legend = FALSE,
                        contour.plot = TRUE,
                        points.plot = TRUE,
                        raster.surf = TRUE,
```



(a) (b) (c)

Figure 1: (a) Contours selected–"A" and "B", marked in yellow, are closed curves and "C", marked in red, is an open curve. (b) 3–D plot of the planar curves collected over time. (c) Triangulation of the *wombling surfaces* associated with the curves using $n_\omega = n_v = 10$.

```
                              extend = TRUE)
  x = raster::rasterToContour(raster.obj, nlevel = 20)
  x.levels <- as.numeric(as.character(x$level))

  wb.1[[j]] = subset(x, level == id.levels[j,1]) # surface A
  wb.2[[j]] = subset(x, level == id.levels[j,2]) # surface B
  wb.0[[j]] = subset(x, level == 0) # surface C

  lines(wb.1[[j]]@lines[[1]]@Lines[[id.lines[j,1]]]@coords,
        lwd = 5, col = "yellow")
  lines(wb.2[[j]]@lines[[1]]@Lines[[id.lines[j,2]]]@coords,
        lwd = 5, col = "yellow")
  lines(wb.0[[j]]@lines[[1]]@Lines[[id.lines.0[j]]]@coords,
        lwd = 5, col = "red")

}
```

2 Creating a Partition

The following code creates Figure 1(b).

```
surf.womb.0 = partition_curve(curve.list = wb.0)
surf.womb.1 = partition_curve(curve.list = wb.1)
surf.womb.2 = partition_curve(curve.list = wb.2)
```

```
require(rgl)

open3d()
bg3d(color = "white")
plot3d(x = surf.womb.0[,1], y = surf.womb.0[,2], z = surf.womb.0[,3],
lwd = 2, xlab = "x", ylab  = "y", zlab = "t", ann = FALSE, axes = FALSE)
box3d()
grid3d(side = c("x", "y", "z"))
rglwidget()
```

3 Triangulating the Surface

The following code creates Figure 1(c).

```
surf.trid.0 = surf_triangulate(curves_part = surf.womb.0); rglwidget()
surf.trid.1 = surf_triangulate(curves_part = surf.womb.1); rglwidget()
surf.trid.2 = surf_triangulate(curves_part = surf.womb.2); rglwidget()

wombling.df.0 = data.frame(do.call(rbind, surf.trid.0$wombling.df))
nrow(wombling.df)
wombling.df.1 = data.frame(do.call(rbind, surf.trid.1$wombling.df))
nrow(wombling.df)
wombling.df.2 = data.frame(do.call(rbind, surf.trid.2$wombling.df))
nrow(wombling.df)
```

4 Surface Wombling

```
WM.obj.0 = sptwombling(model = results[[2]], wombling.df = wombling.df.0)
WM.obj.1 = sptwombling(model = results[[2]], wombling.df = wombling.df.1)
WM.obj.2 = sptwombling(model = results[[2]], wombling.df = wombling.df.2)

WM.obj.0$time["elapsed"]/3600 # 28hrs
WM.obj.1$time["elapsed"]/3600 # 28hrs
WM.obj.2$time["elapsed"]/3600 # 28hrs

############################
# Overall Wombling Measure #
############################
```

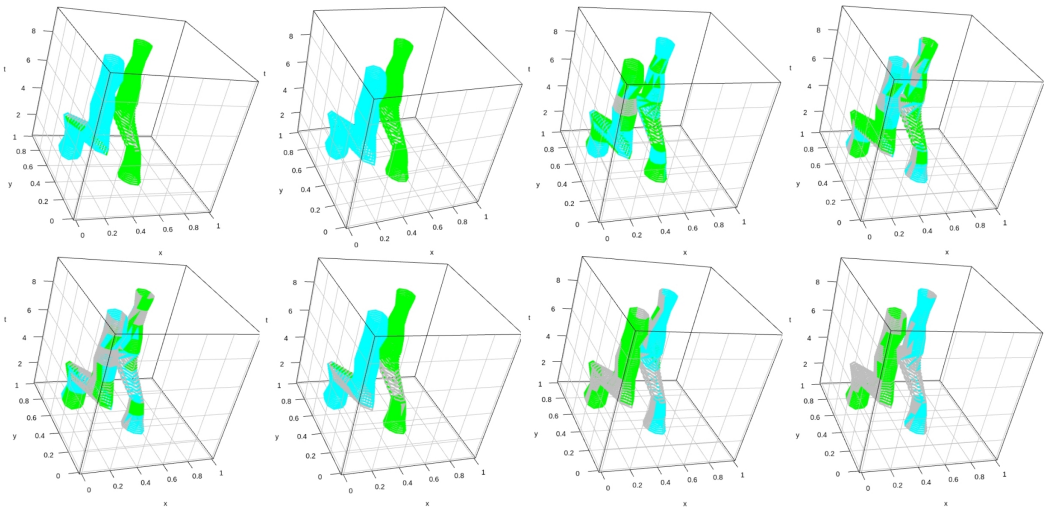


Figure 2: Wombling Surfaces A and B

```
round(do.call(rbind, lapply(WM.obj.0$WM, function(x){
  apply(x[,-4], 2, sum)
})), 6)

############################
# Wombling measures by Time #
############################
WM.bytime = lapply(WM.obj.0$WM, function(x){
  round(rbind(t12 = apply(x[1:18,-4], 2, sum),
              t23 = apply(x[19:36,-4], 2, sum),
              t34 = apply(x[37:54,-4], 2, sum)), 10)
})
WM.bytime
```

5 Significance Assessment

Significance of spatiotemporal wombling measures for wombling surfaces (a) A, B and (b) C (see Figure 1). For Figures 2 and 3 (Row 1: left to right) spatial gradient, spatial curvature, temporal gradient, spatial-temporal gradient, (Row 2: left to right) temporal gradient in spatial curvature, temporal curvature, temporal curvature in spatial gradient, spatial-temporal curvature. Regions marked `green` (`cyan`) indicate positive (negative) significance while `grey` indicates no significance.

References

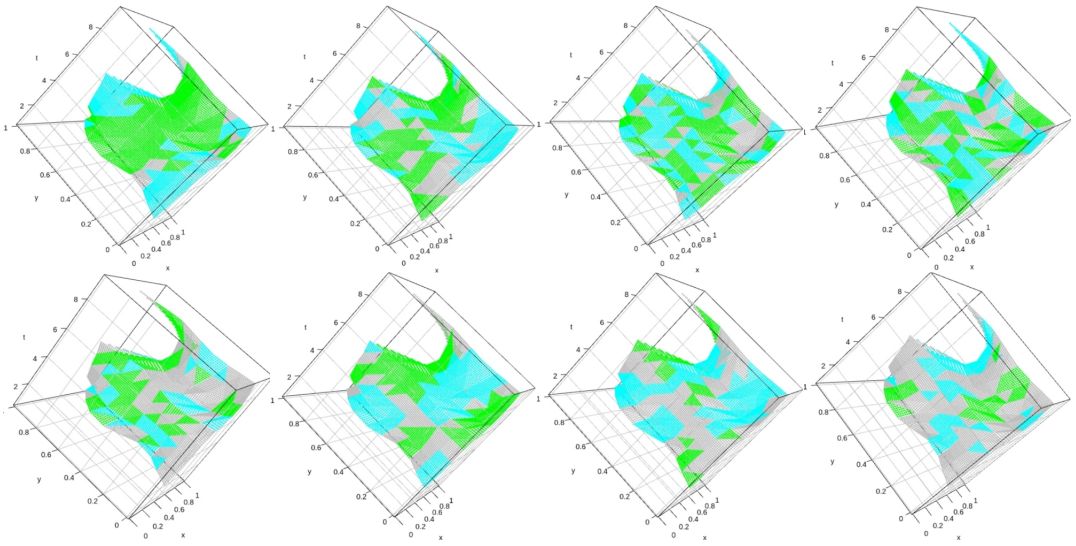


Figure 3: Wombling Surface C