

Untitled.R

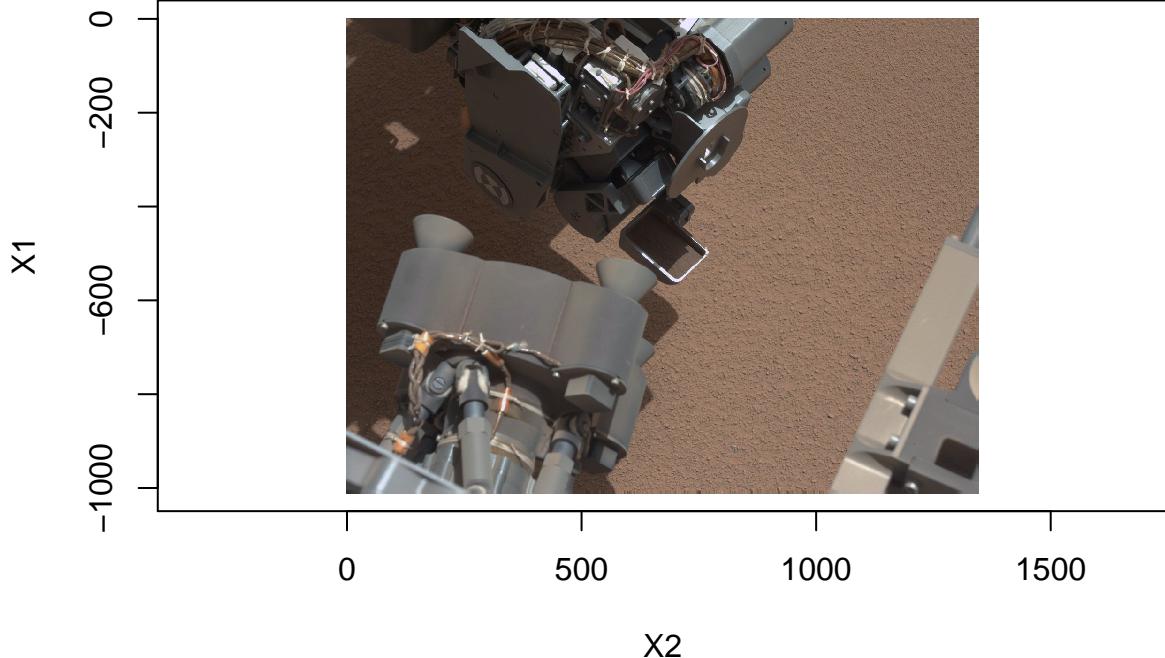
Ashish

Fri Sep 26 22:33:59 2014

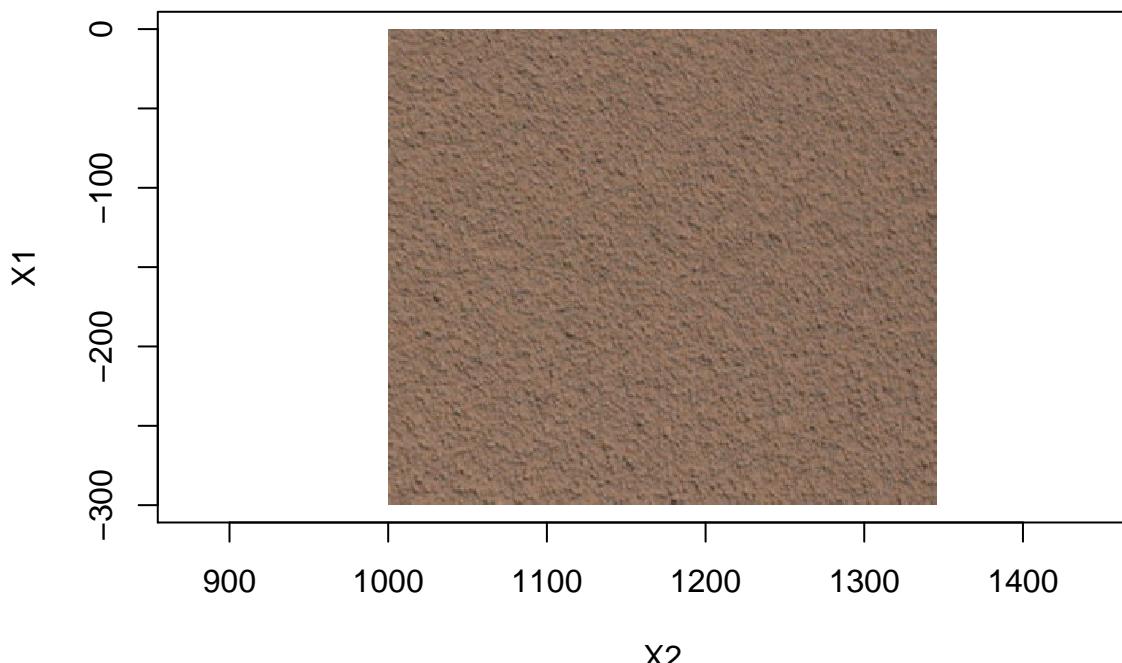
```
library(jpeg)
library(reshape)
library(ggplot2)
imageLoader <- function(url){ # This function takes a URL, and generates a data.frame with pixel location
  # Download to disk, load
  download.file(url, "tempPicture.jpg", mode = "wb") # Stash image locally
  readImage <- readJPEG("tempPicture.jpg")

  longImage <- melt(readImage)
  rgbImage <- reshape(longImage, timevar = "X3",
                       idvar = c("X1", "X2"), direction = "wide")
  rgbImage$X1 <- -rgbImage$X1
  return(rgbImage)
}

# Load NASA's image
rgbImage <- imageLoader("http://www.nasa.gov/images/content/694811main_pia16225-43_full.jpg")
with(rgbImage, plot(X2, X1, col = rgb(rgbImage[, 3:5]), asp = 1, pch = "."))
```



```
# Look at a region of only sand
sandOnly <- with(rgbImage, rgbImage[X2 > 1000 & X1 > -300, ])
with(sandOnly, plot(X2, X1, col = rgb(sandOnly[, 3:5]), asp = 1, pch = "."))
```

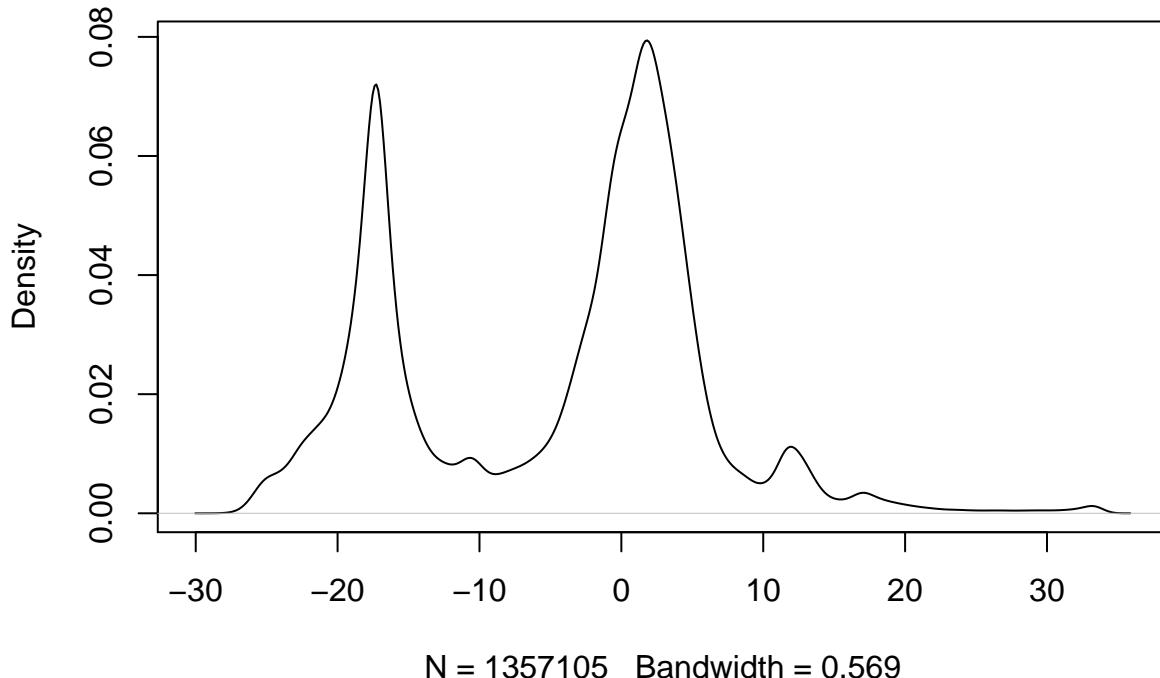


```

sandMean <- apply(sandOnly[, 3:5], 2, mean) # Find typical sand color values
sandSD <- apply(sandOnly[, 3:5], 2, sd)
colorZ <- sweep(rgbImage[, 3:5], 2, sandMean, "-") # How much do all colors
colorZ <- sweep(colorZ, 2, sandSD, "/") # deviate from sand colors?
plot(density(rowSums(colorZ)))

```

density.default(x = rowSums(colorZ))



```
# Plot a "binarized" image, based on color Z-scores
with(rgbImage, plot(X2, X1, col = rgb(colorZ>4), asp = 1, pch = "."))
points(830, -920, cex = 4, col = "RED")
```

