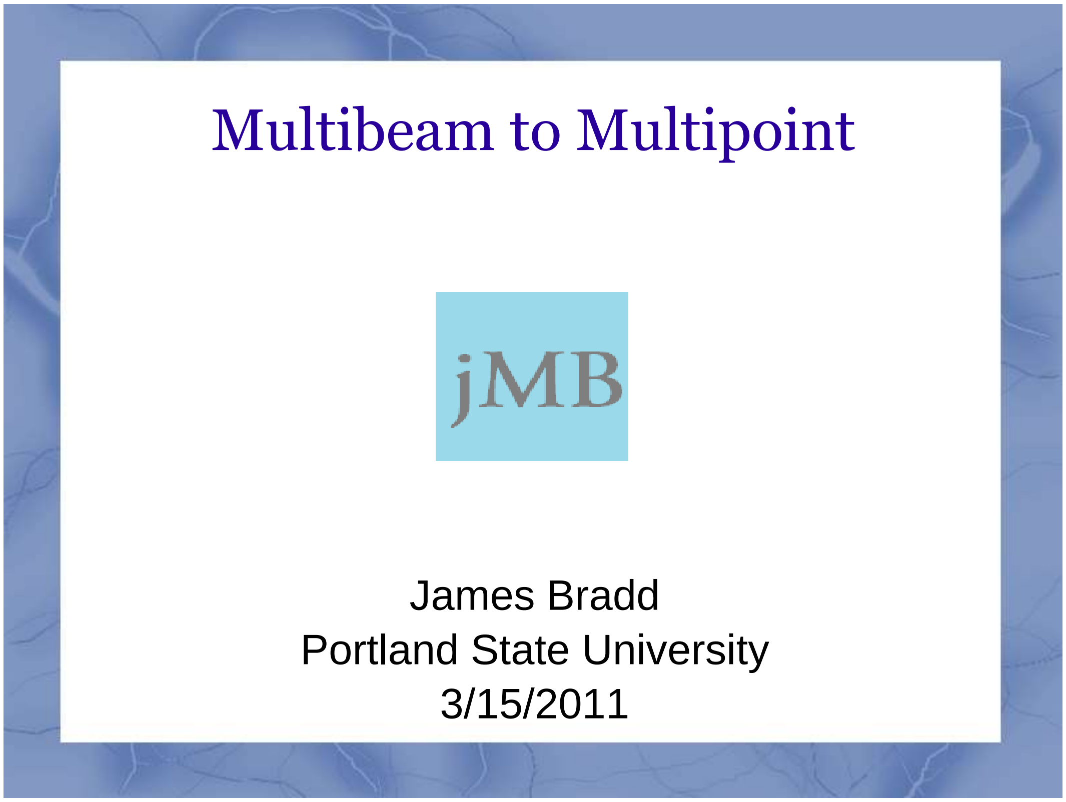
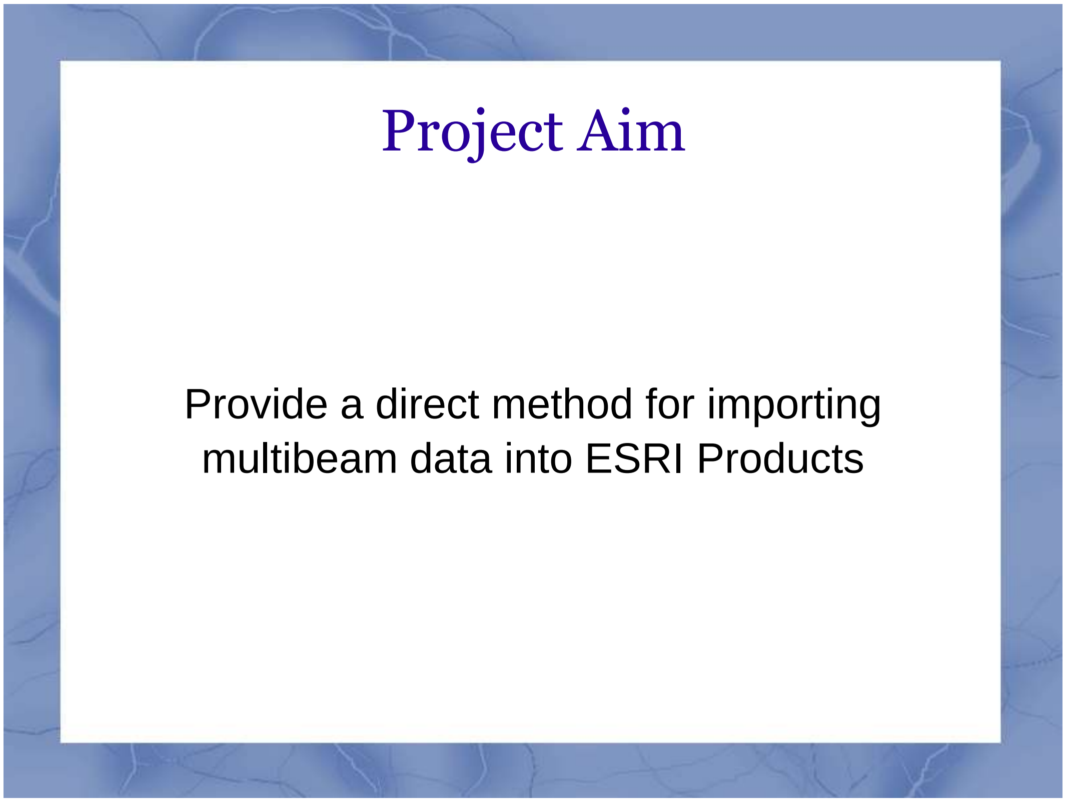


# Multibeam to Multipoint

jMB

James Bradd
Portland State University
3/15/2011

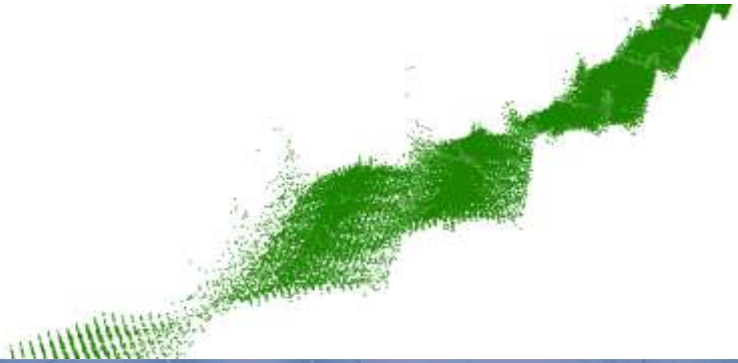


## Project Aim

Provide a direct method for importing multibeam data into ESRI Products

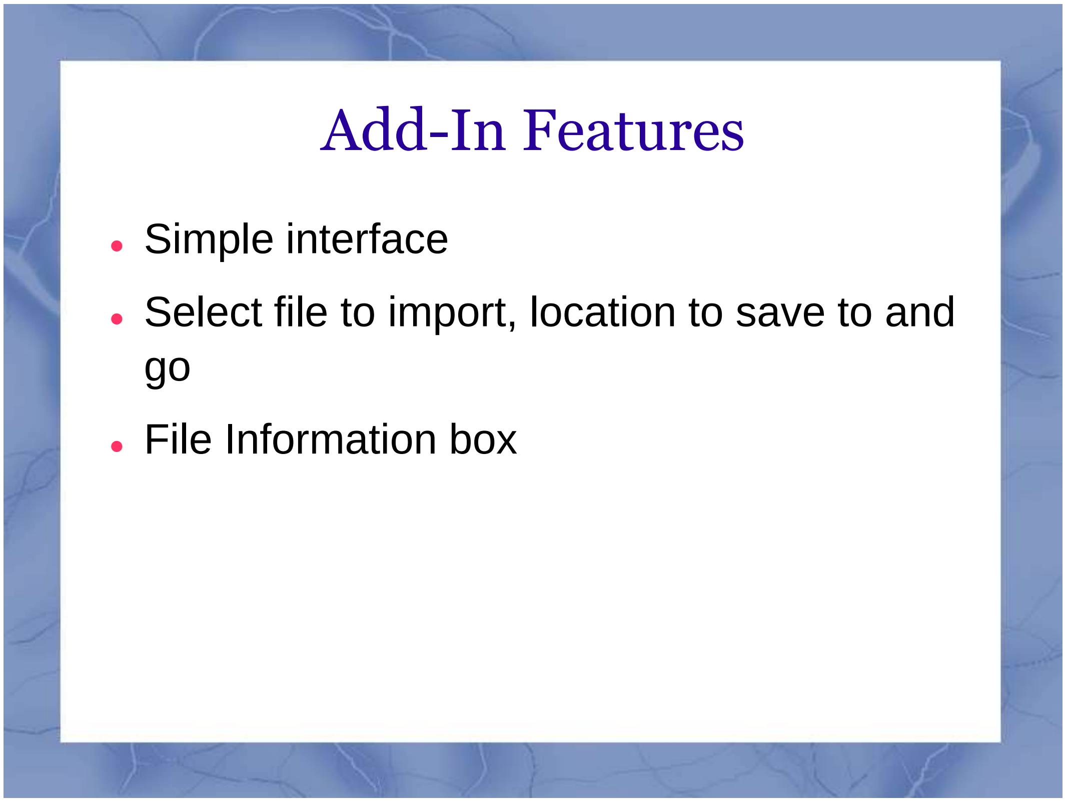
# Mutibeam?

Working Definition: The LiDAR of the Sea



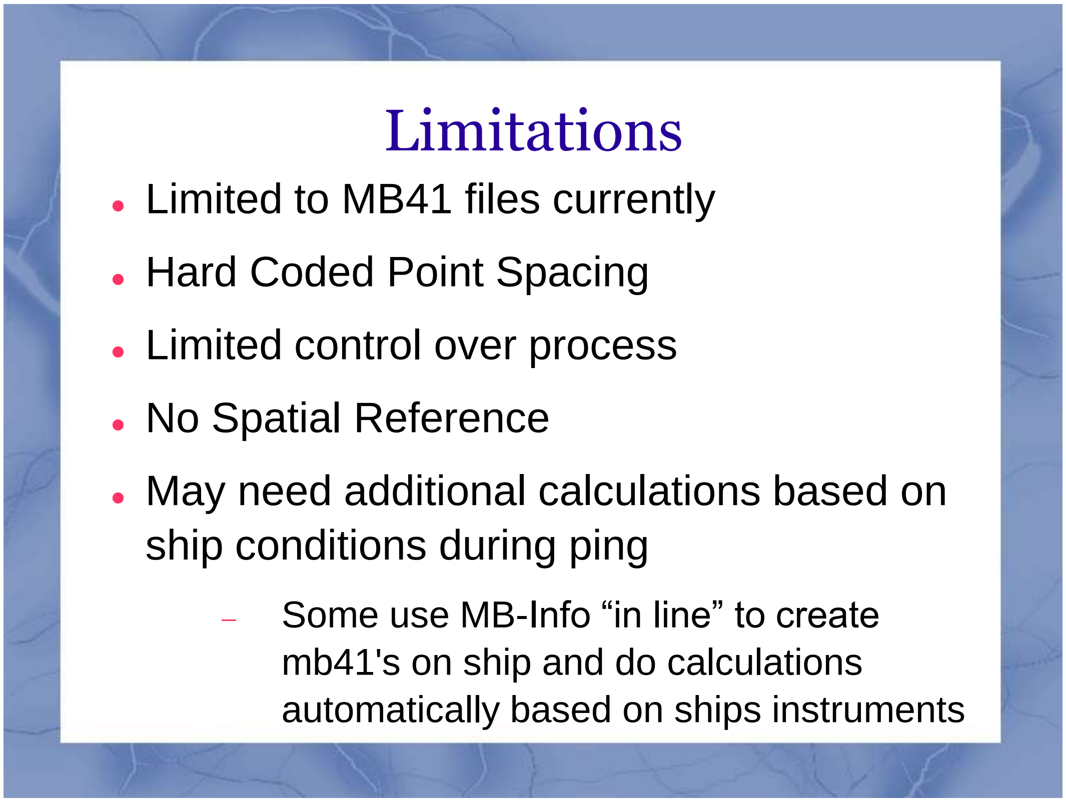
# Multibeam Processing Overview

- Only free way I could find is with MB-System on a UNIX or Linux based machine
- No direct method for getting into ESRI products
- Costly and time consuming to run virtual linux if you only have a windows machine



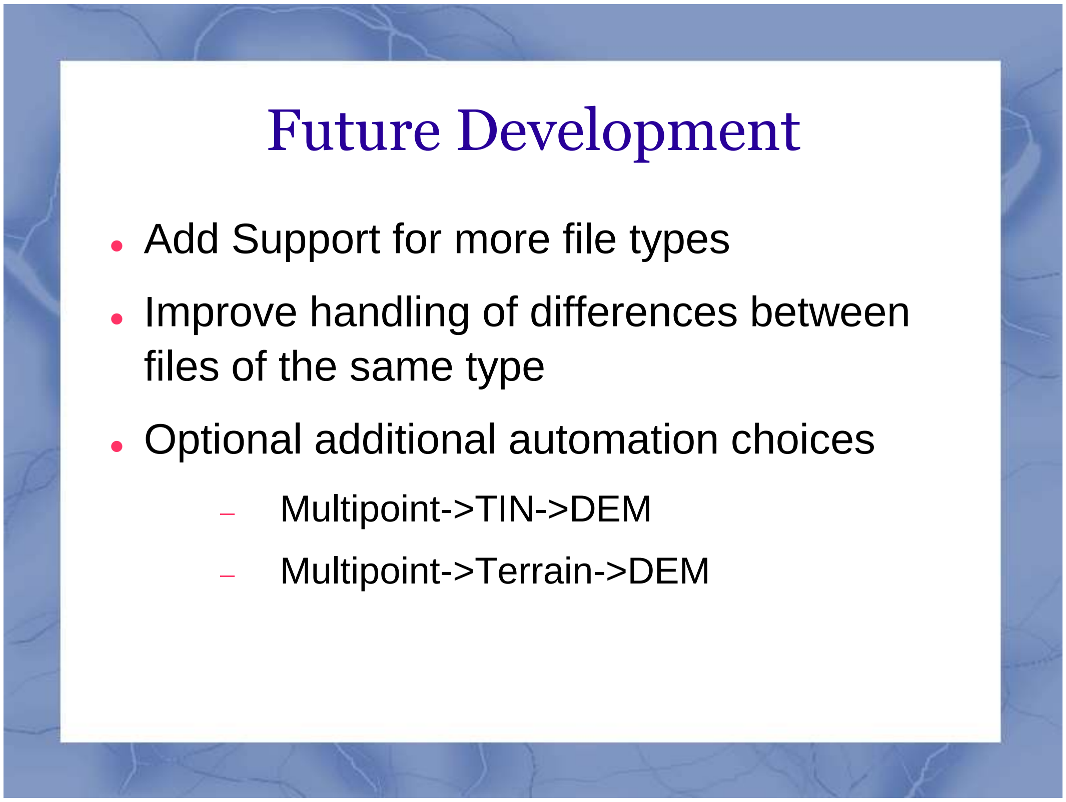
# Add-In Features

- Simple interface
- Select file to import, location to save to and go
- File Information box



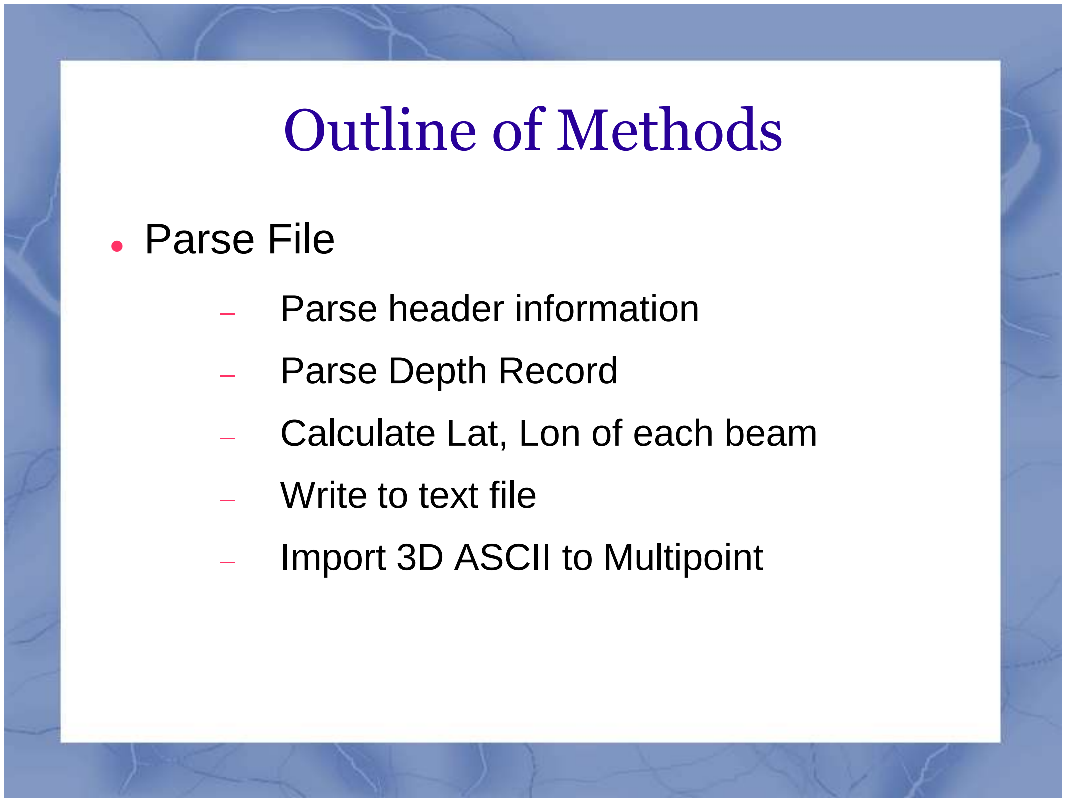
# Limitations

- Limited to MB41 files currently
- Hard Coded Point Spacing
- Limited control over process
- No Spatial Reference
- May need additional calculations based on ship conditions during ping
  - Some use MB-Info “in line” to create mb41's on ship and do calculations automatically based on ships instruments



## Future Development

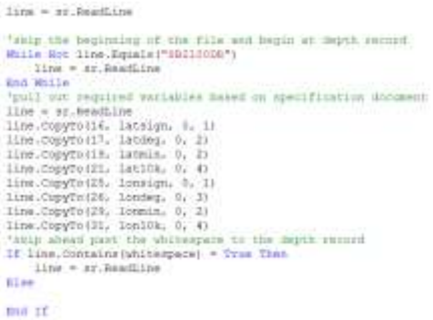
- Add Support for more file types
- Improve handling of differences between files of the same type
- Optional additional automation choices
  - Multipoint->TIN->DEM
  - Multipoint->Terrain->DEM



## Outline of Methods

- Parse File
  - Parse header information
  - Parse Depth Record
  - Calculate Lat, Lon of each beam
  - Write to text file
  - Import 3D ASCII to Multipoint

# Parse Depth Record Header



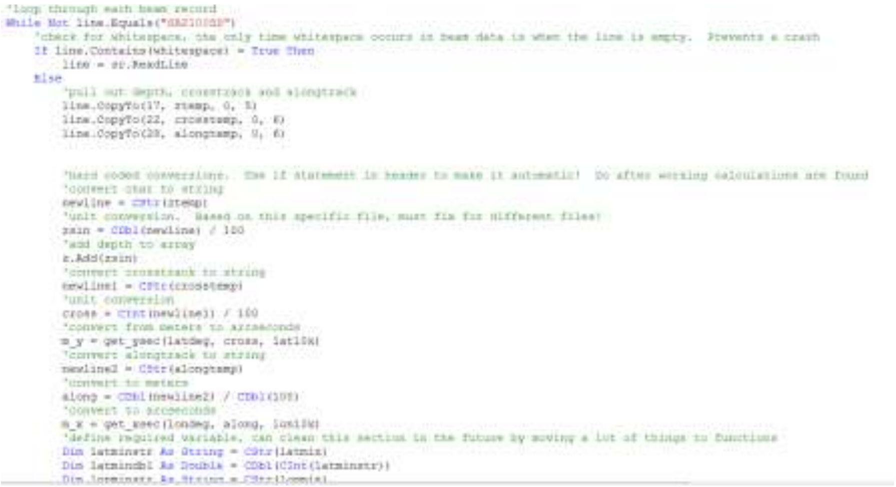
```
line = sr.ReadLine

'skip the beginning of the file and begin at depth record
While Not line.Equals("SB2130DD")
    line = sr.ReadLine
End While
'pull out required variables based on specification document
line = sr.ReadLine
line.CopyTo(16, latsign, 0, 1)
line.CopyTo(17, latdeg, 0, 2)
line.CopyTo(18, latmin, 0, 2)
line.CopyTo(21, lat10k, 0, 4)
line.CopyTo(25, lonsign, 0, 1)
line.CopyTo(26, londeg, 0, 3)
line.CopyTo(29, lonmin, 0, 2)
line.CopyTo(31, lon10k, 0, 4)
'skip ahead past the whitespace to the depth record
If line.Contains(whitespace) = True Then
    line = sr.ReadLine
Else

End If
```

- Combine FileStream and StreamReader
- Read line by line for keywords (stopping points)
- Pull out numbers given location on specification sheet

# Parse Per/Beam Records

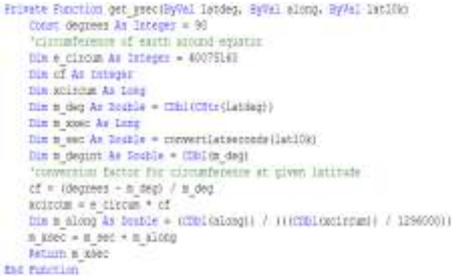


```
'loop through each beam record
While Not line.Equals("SB2130BP")
    'check for whitespace, the only time whitespace occurs in beam data is when the line is empty.  Prevents a crash
    If line.Contains(whitespace) = True Then
        line = sr.ReadLine
    Else
        'pull out depth, crosstrack and alongtrack
        line.CopyTo(17, ztemp, 0, 5)
        line.CopyTo(22, crosstemp, 0, 6)
        line.CopyTo(28, alongtemp, 0, 6)

        'Hard coded conversions.  Use if statement in header to make it automatic!  Do after working calculations are found
        'convert char to string
        newline = CStr(ztemp)
        'unit conversion.  Based on this specific file, must fix for different files!
        zmin = CDbl(newline) / 100
        'add depth to array
        z.Add(zmin)
        'convert crosstrack to string
        newline1 = CStr(crosstemp)
        'unit conversion
        cross = CInt(newline1) / 100
        'convert from meters to arcseconds
        m_y = get_ysec(latdeg, cross, lat10k)
        'convert alongtrack to string
        newline2 = CStr(alongtemp)
        'convert to meters
        along = CDbl(newline2) / CDbl(100)
        'convert to arcseconds
        m_x = get_xsec(londeg, along, lon10k)
        'define required variable, can clean this section in the future by moving a lot of things to functions
        Dim latminstr As String = CStr(latmin)
        Dim latmindbl As Double = CDbl(CInt(latminstr))
        Dim lonminstr As String = CStr(lonmin)
```

## Calculate Beam Location

- Given x,y offset in meters
  - Convert to arc seconds
  - Add to origina
  - Convert to decimal



```
Private Function get_xsec(ByVal latdeg, ByVal along, ByVal lat10k)
    Const degrees As Integer = 90
    'circumference of earth around equator
    Dim e_circum As Integer = 40075160
    Dim cf As Integer
    Dim xcircum As Long
    Dim m_deg As Double = CDbl(CStr(latdeg))
    Dim m_xsec As Long
    Dim m_sec As Double = convertlatseconds(lat10k)
    Dim m_degint As Double = CDbl(m_deg)
    'conversion factor for circumference at given latitude
    cf = (degrees - m_deg) / m_deg
    xcircum = e_circum * cf
    Dim m_along As Double = (CDbl(along)) / (((CDbl(xcircum)) / 1296000))
    m_xsec = m_sec + m_along
    Return m_xsec
End Function
```

## Getting Data Into Map

- Write X,Y,Z to temp file
- Use a geoproccessor object to execute 3DASCIItoFeatureClass
- Save feature class to user defined location

# Uses

- Create DEM's
- DEM Derived products
- Create giant filesizes
- High end visuals when combined with LiDAR
- ???
  - The more tools we can provide, the better

# Questions?