

CS171 Project 2 Process Book

Visualizing Official Development Assistance (ODA) in 2011

Kevin Sun & Albert Young

April 8, 2013

TABLE OF CONTENTS

Project Proposal.....	2
Project title: A Geographic Visualization of Official Development Aid (ODA) Data	2
Project members:	2
Research questions and hypotheses:	2
Motivation:	2
Data:	2
Visualization:	3
Sketches:	3
Project Development.....	4
Data Collection and Processing:	4
Geographic Coordinates:	4
Official development assistance (ODA) Data:	4
Visualizing Data	5
Technology:	5
Map Shading:	5
Donor Flow Lines	5
Country Selection	6
Dynamic Tooltips.....	7
Flow Highlighting:	7
Pie Charts:	7
Reset View and Current Country Focus:	8
Animating Data.....	8
Animating Aid Flow Lines.....	9
Filtering Data:	9
Line Flow Filter:	9
Design Justifications:.....	10
Mockup selection:	10
Color Scheme:	10
Page Layout:	11
Observations and analysis.....	11
Project Summary and Evaluation:	13
Appendix: Sketches.....	14

PROJECT PROPOSAL

PROJECT TITLE: VISUALIZING OFFICIAL DEVELOPMENT AID (ODA) DATA IN 2011

PROJECT MEMBERS:

- **Kevin Sun** (ksun01@college.harvard.edu)
- **Albert Young** (ayoung01@college.harvard.edu)

RESEARCH QUESTIONS AND HYPOTHESES:

- Where (which countries) does aid funding come from?
 - Which country provides the most aid funding? Which are the top countries in terms of assistance provided?
- For what purposes and/or to which sectors do aid flows go?
 - Are these proportions related to the country receiving the aid data?
 - Are there countries that prioritize certain forms of aid over others?
- How does aid funding change over time with regards to amount and composition?

MOTIVATION:

- Our research questions provide a platform to examine aid data and educate the public about international development efforts they pay for. In that regard, we are motivated to visualize aid data because we believe doing so can create a worthy sense of accountability towards aid organizations.
- There is a clear division between Development Assistance Committee (**DAC**) countries and non-DAC countries. DAC countries are OECD member countries that are identified as the world's major donor countries. We can examine general trends for DAC countries in various aid sectors between countries and across the years, making aid data readily accessible.
- In addition, we postulate that having a means to visualize aid data could lead to better (or smarter) distribution of aid resources moving forward. For example, if country A provides public sector aid to country B, then country Y does not need to (and can focus country Y's aid efforts elsewhere).

DATA:

- The data that we will be using will be from the following resources both of which offer downloadable files of aid funding numbers that will be directly visualized by the project:
 - The Organization of Economic Cooperation and Development ([link](#))
 - This information is from the Organization of Economic Co-Operation and development, and contains chronological and sector-specific data.
 - Aiddata.org ([link](#))

- This website compiles data from different ‘donor’ organizations, and has a searchable database that could fill in missing dimensions from OECD data, such as country to country flows of aid information.

VISUALIZATION:

We would like to display our data to facilitate comparison between countries and categories:

- Ideas that we aim to implement:
 - Create a world map that allows filters based on aid-giving and aid-receiving countries (potentially incorporating Google Maps or this [azimuthal projection](#))
 - Different colors/shading of countries will encode different values based on the filters selected
 - Clicking on a country will bring up a dashboard for the country with more specific information about the quantity of and what types of aid it is giving/receiving
 - Rank the top aid-giving and aid-receiving countries
 - Encode net donor information with color.
- Ideas that would be ideally implemented:
 - Allow the user to filter the map/dashboards based on time
 - Allow the user to ‘play’ the chronological development of aid data
 - Create interactive links that will allow the user to access the specific aid instance that data was derived from
 - Visualize the flow of aid data between countries using graphical elements and/or animations (e.g. lines connecting the countries such that the weight of the line is representative of the flow between the countries)
 - Allow the map to be interactive (pannable/zoomable)

SKETCHES:

In this file we have attached the following three sketches:

- 1 **Side Panel with Country Flows:** Upon clicking a country, a side panel with various charts (sector breakdown and top country flows in bar chart format) will update to show country-specific information. Lines between the selected country and other donor/recipient countries will highlight the quantity of aid flows, with line width as representative of the amount of data provided/received.
- 2 **Large Panel Overlay:** Upon clicking a country, a large dashboard will appear with more country-specific information encoded in a number of graphical elements (chronological bar chart, sector breakdown pie chart, country image).
- 3 **Line Diagram:** This sketch uses line width as a proxy for donor flows. By aligning donor countries on one side and recipient countries on the other, users can determine the aid cash flow relationships between the two sides. The dynamically-adjusted panel on the bottom provides a sector breakdown.

PROJECT DEVELOPMENT

DATA COLLECTION AND PROCESSING:

GEOGRAPHIC COORDINATES:

- Data was pulled from an HTML table located [here](#):
- Using Excel and index-matching, the tables countries were converted to two-digit country codes.
- Using Excel's 'text to column' feature and formulas, each coordinate in the form of '45 00 N' was converted to a simple decimal value (coordinates with S or E were multiplied by negative one)
- Finally, the data was converted using a Python script to JSON format.
- The JSON format was included in index.html as `textarea` with `"id=countrycenters,"` then parsed into an associative array.

OFFICIAL DEVELOPMENT ASSISTANCE (ODA) DATA:

- From OECD's wide offering of aid statistics, we first needed to decide what data was most interesting and which could be visualized most effectively
- The following Excel files were downloaded from <http://www.oecd.org/statistics/> from the Aid (development) section
 - Aid by Major Purposes (commitments)
 - Net ODA by Income Group
 - Net ODA by Region
 - Net Official Development Assistance: Development Aid
- The excel files were manually filtered for the correct year and merged, outputted in CSV format, and cleaned with Google Refine (e.g. to convert text to numeric data types).
- In order to determine the country-to-country specific Net ODA flows, we queried <http://stats.oecd.org/> for each individual DAC country's Net ODA to each non-DAC country and downloaded and manually merged each of the resulting Excel files.
- Because data was only consistently available in all categories for 2011, we decided to filter all data by only that year.
- Finally, the data was converted using a Python script to JSON format and embedded directly into index.html.
- At first, we tried to include the JSON files separately and to process them using `jQuery.getJSON()`, but we ran into trouble with our local files due to the same origin policy. It was easier to include the JSON data directly inside `textareas` in `index.html`.

VISUALIZING DATA

TECHNOLOGY:

- We chose to use the **jVectorMap** Javascript package (<http://jvectormap.com/>) to provide a basic map framework. We chose jVectorMap over other candidates like Google Maps and DataMaps.js for several reasons
 - jVectorMap natively supports features such as panning and zooming, tooltips, highlighting of countries upon mouseover, and color-coding of countries based on data input
 - jVectorMap has offers a detailed documentation of its functions, which helped when we had to make changes to certain library functions
 - We did not use Google Maps because it does not not integrate as well with data and is too detail-rich for our purposes
 - We did not use DataMaps because it offered fewer features that jVectorMap (in particular zooming and panning, which was essential to viewing flows for smaller countries)
- We used **D3.js** to construct the pie charts in our visualization because we had prior experience with D3.js from class and it offered simple tools for creating and manipulating pie chart elements in the DOM. In addition, D3's transition capabilities were used to animate the flow lines.

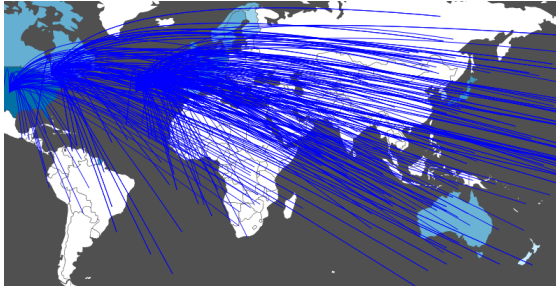
MAP SHADING:

The map shading scale used for our visualization proved nontrivial because of the two datasets we had to encode in the color of the map (positive donor information and negative recipient information). We followed these steps:

- Because jVectorMap only accepts one dataset as shading information, we combine the dataset, multiplying all recipient values by negative 1.
- The jVectorMap color scaling process only uses one color scale, which would be problematic when we needed to encode values along two different color schemes.
- To solve this problem, we modified the jVectorMap numeric scale library and hardcoded in the appropriate minimum maximum values such that there were two distinct datasets [0, maxdonor] and [0, maxrecipient].
- Finally, I edited the jVectorMap scaling library to use two distinct scales based on the positive/negative sign of the value being passed through.
- The end results is varying shades of blue for donor countries and varying shades of red for recipient countries, with the more saturated/darker colors expressing greater magnitude.

DONOR FLOW LINES

- Drawing donor flow lines relied on appending additional 'path' elements to the SVG container created by the jVectorMap framework.

- The function `drawLine` was created to facilitate the drawing of such lines, taking in geographic coordinates and a width, appending a SVG element (a Bezier curve, for aesthetic reasons) with the appropriate width.
 - The geographic coordinates are passed through a `latLngToPoint` function in the `jVectorMap` framework that projects the geographic points to the map.
 - Because straight looked unappealing, we decided to make the paths Bezier curves. More specifically, we used quadratic Bezier curves
 - The 'third point' selected was used to create downwards-concavity for donor countries and 'upwards' concavity for recipient countries.
- After `drawLine` was created, `drawCountryFlows` was created. This function takes in a country code and draws all the flows to/from the country.
 - The function determines whether or not the country is a recipient or donor country.
 - Next, the function pulls an associative array from `donationadata`, which functions as a dictionary of associative arrays.
 - Finally, the function iterates over each element in the "flow array" for the selected country using a `for each` loop. The geographic coordinates are pulled in from the associative array `geographiccenters` and passed to `drawLine`.
- Finally, the donor flow lines were altered to be cleared and redrawn with each map navigation movement.
 - Without making this alteration, the map looking liked the following:
 
 - To make this alteration, the lines were redrawn and `drawCountryFlows` was called on stored `currentcountry` every time the `jVectorMap` handlers of `onViewportChange` were processed.
- Code was modified using if statements to create similar lines for recipient countries.
 - To differentiate the lines, the recipient lines have a positive-y-shifted quadratic bezier curve and are colored differently in red.
- The lines were changed to be of variable color (for positive/negative and donor/recipient flows) as well as different width, to illustrate scale.
 - Note that countries can receive a negative amount of aid (if they have to pay aid back).

COUNTRY SELECTION

- The next major step of our visualization was to be able to select each country and retrieve information about the flows to and from that country to specific countries, incorporating **details-on-demand**.
- The map was modified to allow for the selection of **one** region at a time by clicking.

- This selection was achieved by modifying `regionsSelectableOne` of the `jVectorMap` framework.
- The effect of selecting a country provides a filtering mechanism.
 - The tooltips change to visualize aid just for that one country.
 - The lines change to visualize for one country.
- These filtering mechanisms were constructed through the `donationdata` and `recipientdata` variables, which are associative arrays of associative arrays.
 - By testing whether or not a country code is 'defined' in either variable, we can determine whether or not a country is a 'donor' or recipient'
- Because `jVectorMap` only includes a subset of the world's countries, we had to ensure that countries which were in our dataset (but not in the actual map) were not visualized using control logic.

DYNAMIC TOOLTIPS

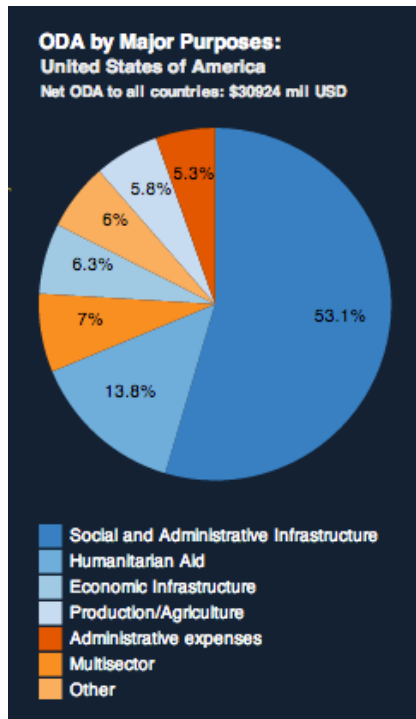
- The use of tooltips in our visualization incorporates details-on-demand and go beyond just a simple display of data. Depending on the context, the tooltip displays different sets of data:
 - When no country is selected, the tooltip displays the total amount of flow donated or received. This information matches the visualized data. For donor countries, the tooltip was modified by using the `jVectorMap` handler `onRegionLabelShow` to display Net ODA provided.
 - For recipient countries, the same handler displays Net ODA received.

FLOW HIGHLIGHTING:

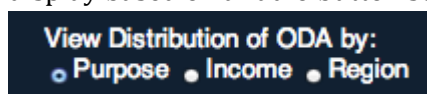
- Following our country selection, we decided to highlight certain 'aid flows' on mouseover connecting the receiving and giving countries, supporting **linking & brushing**.
- For example, if the United States was selected and the user mouseovers South Sudan, the aid flow line from the United States to South Sudan would become highlighted.
- This was first accomplished through drawing another differently-colored path over the old path, but this solution was aesthetically unappealing (as drawing the path to exactly cover the old one proved impossible).
 - To solve this problem, we redrew the lines using `clearLines` and `redraw` along with a special inclusion of a highlighted route.

PIE CHARTS:

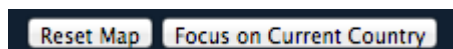
- We incorporate **linking & brushing** and **details-on-demand** with the pie charts, which display data in a separate visualization based on the current selected country.



- When a DAC country is selected, we use D3.js to create a pie chart showing the distribution of ODA by major purposes, by income of the recipient countries, and by region.
- Each slice displays the percentage value (rounded to one decimal place) that will fit comfortably on the slice, and the color encodings are indicated by a separate legend below the pie chart
- We adapted the simple pie chart example at <https://gist.github.com/enjalot/1203641> to incorporate our data.
- We incorporate **drill-down/filter capability** by allowing the user to switch the data display based on a radio button selection:



RESET VIEW AND CURRENT COUNTRY FOCUS:



- We include the 'Reset Map' and 'Focus on Current Country' buttons to help the user navigate the map, incorporating **details-on-demand**.
- When the 'Reset Map' button is clicked, we reset the map view to the full zoomed-out view, clear all currently drawn lines (with `clearLines`), and de-select the current country (by setting `currentcountry = undefined`).
- When the 'Focus on Current Country' button is clicked, we zoom in to the currently selected country (using `$('#map1').vectorMap('set', 'focus', currentcountry);`).

ANIMATING DATA

To increase our visualization's visual appeal and also to offer more guidance on which direction aid was flowing, we decided to include animations for our project:

ANIMATING AID FLOW LINES

Animating aid flow lines proved to be a nontrivial task. Based on Internet research, our first approach was to add an `<animate />` element to the SVG, which did not function appropriately for us. The next approach was to use D3 in some capacity. After deciding that `D3.svg.line()` was unnecessarily complicated, we came to the conclusion to use a hack using the stroke-dash array function and D3's `transition` function.

- The animation was constructed primarily through the `redraw` function, though not all redraws are animated.
 - We left in the function to build non-animated visualizations, mainly for rescaling purposes.
 - `redraw` now takes the variable `animate` which specifies whether or not to animate.

The animation contains the following features:

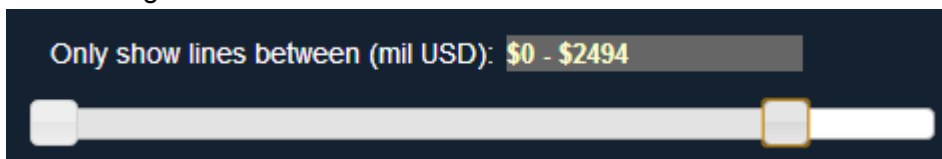
- *Opposite direction for donor/recipient countries*
 - For a selected donor country, the lines use linear interpolation and go from the donor country [selected] to the recipient countries.
 - For a selected recipient country, the lines also linear interpolation but instead go from the donor countries to the [selected] recipient country.
 - This was technically accomplished by modifying the 'd' attribute of the path.
- *Dynamic distance calculation*
 - The animations are timed such that the lines all arrive at their distance at the same time. The animation speed for each line is different.
- *Timing:*
 - Based on trial-and-error, we chose the animation time of 700ms

FILTERING DATA:

By construction, our visualization includes a number of 'filtering' mechanisms, including the ability to select one particular country, show the sector breakdown of that country, and visualize flows of that country. In order to increase functionality and **drill-down/filter capability**, we also decided to add a filter for the lines being drawn:

LINE FLOW FILTER:

- The filtering mechanism is as follows:



- Filter functionality:

- The filter can be adjusted either using the slider or the form field, which is an editable field that is automatically parsed into values.
 - The form field and slider sync with one another.
 - Any change to the form field/slider calls `redraw`, which redraws all the lines on the map with animation.
- The filter was based in jQuery, and we decided on a range filter because of aesthetic design and functionality.

DESIGN JUSTIFICATIONS:

MOCKUP SELECTION:

- We decided to select the first sketch concept as the basis for our visualization design. The lines drawn between countries is the most intuitive and effective way to encode aid flows between countries encourages interactivity with highlights and animations. As in the sketch, we ended up using a separate display panel for the ODA distribution data but decided a pie chart was a more appropriate form of visualization.
- The second sketch concept was abandoned because the large overlay panel would obscure too much of the map and make navigation difficult. We also underestimated the space that each of the smaller visualizations would take.
- The third sketch concept was abandoned because visualizing lines between lists of countries was less informative than visualizing lines on a map. Also, the ability to navigate and explore the map would be lost.

COLOR SCHEME:

- Background:
 - We used a deep Prussian blue for the background color of the page in order to contrast with the brighter elements of the page--the data being highlighted.
 - The background color has low saturation in order to give it an unimposing presence
- Map:
 - DAC countries were rendered with different shades of blue, and non-DAC were rendered with different shades of red. We chose these colors because they lie on far sides of the color spectrum and allow easy distinction between DAC and non-DAC countries.
 - Countries that were missing data were rendered in white. The user should easily associate the lack of color with missing data values.
 - When a country is selected, its color changes to a bright yellow-orange, which gives it the most contrast against the background and allows the user to easily identify the current selection.
 - Upon mouseover, the shade of a country deepens enough to provide feedback to the user that a new country is being moused over. This is the unmodified default state of the jVectorMap example we built off of.
- Pie Chart:

- The pie chart uses shades of blue and orange similar to those found on the map, integrating color harmony and making it easier for the user to connect the distinct elements of the page.
- The blue identifies well with the blue shade of the DAC countries, orange is a complementary color to blue which increases contrast between each consecutive slice of the pie.
- Flow Lines:
 - We used a saturated blue-green color to identify outgoing flow lines from DAC countries and a saturated red color to identify incoming flow lines for non-DAC countries. The blue and red colors help the user to identify the flow lines as belonging to DAC or non-DAC countries, respectively, and have sufficient contrast against each other and against the background to be followed easily.
 - Upon mouseover, the flow line is highlighted in white, which provides contrast against other flow lines and underlying countries, but especially helps distinguish the line from the background.
- Tooltips:
 - Tooltips have a dark gray background with a white stroke, a default feature of the jVectorMap example, which we found easy to read and unnecessary to modify.

PAGE LAYOUT:

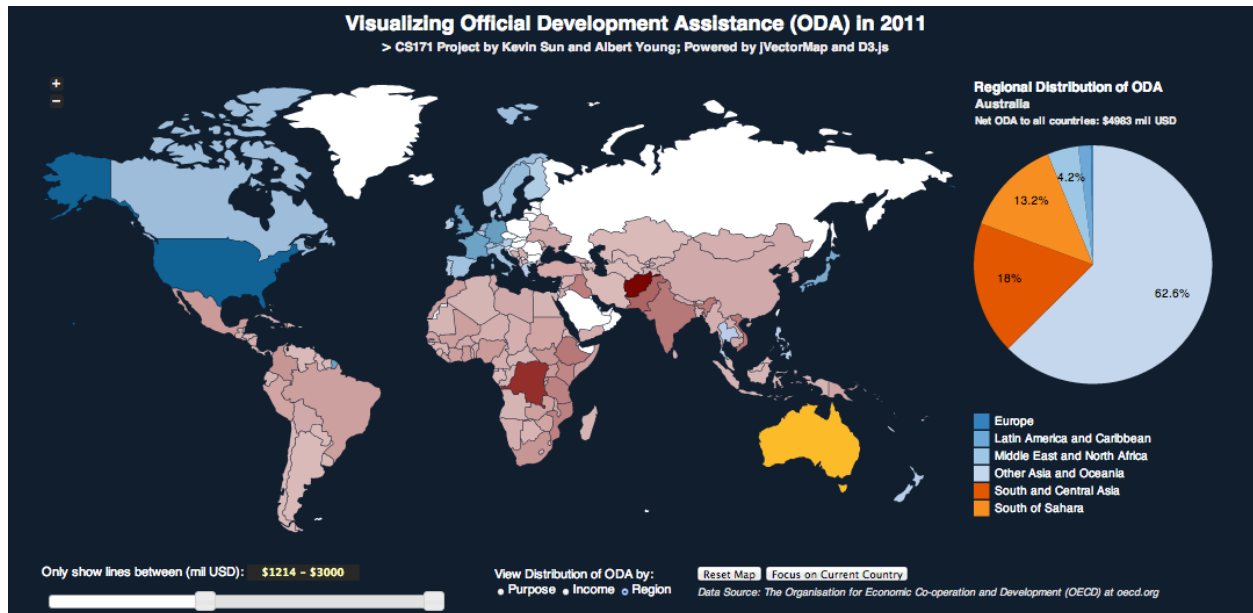
- Resolution:
 - We decided to use a 1366 x 768 size for our visualization because it is now the most popular screen resolution for the web and was appropriate for our layout, based on Internet research (<http://techcrunch.com/2012/04/11/move-over-1024x768-the-most-popular-screen-resolution-on-the-web-is-now-1366x768/>).
- Layout:
 - We tried as much as possible to have all of the data and controls present in a single view without the necessity of scrolling. Originally, we planned to have the three pie charts displayed underneath the map, but there was not enough resolution to contain all of the pie charts and their legends. In the end, we settled on keeping a single large pie chart on the same level as the map, where the data underlying the pie chart can be easily toggled with the radio buttons.

OBSERVATIONS AND ANALYSIS

- From exploring our map data, we made the following observations:
 - The United States is by far the largest donor, with \$30924 million USD in ODA in 2011, and the Democratic Republic of the Congo and Afghanistan are the largest recipients of ODA (\$4248.8 and \$5751.8 million USD, respectively). These countries have the darkest shading on the map.
 - DAC countries are concentrated in North America (U.S., Canada), Western Europe, and Australia and New Zealand. This is no surprise.

- Each DAC country has aid flows between most other non-DAC countries, but there are few flows above \$200 million USD. Most flows are under \$10 million USD. This can be observed by adjusting the flow line filter.
- Many countries are receiving negative Net ODA from Japan (i.e. they are repaying ODA-related loans), such as China, Indonesia, and Thailand. This can be seen by the red lines flowing into Japan. We are not sure of the economic reasoning behind this, but it is an interesting observation.
- Thailand has an interesting role on our map because even though it is a non-DAC country, it has a blue fill which means there is more ODA leaving the country than it receives (mainly because of its large repayment to Japan).
- We are surprised by the lack of ODA data for Russia. Perhaps the figures are not publicly released.
- From exploring our pie chart data, we made the following observations:
 - The majority of ODA by purpose goes to Social and Administrative Infrastructure, with the exception of Italy and Japan (which focuses on 'Other' and Economic Infrastructure, respectively).
 - A surprisingly small portion of ODA from each DAC country is dedicated as Humanitarian Aid (typically <10%).
 - The majority of ODA by Income goes to Least Developed Countries (LDCs), with the exception of Australia, which donates 51.9% of its Net ODA to Low- or Middle-Income Countries.
 - There is more ODA in general directed toward Low- or Middle-Income Countries versus Low-Income Countries. Perhaps there are more countries that receive the former designation.
 - Countries tend to offer more aid locally—Australia and New Zealand offer 62.6% and 76.2% of their ODA, respectively, to 'Other Asia and Oceania', and Japan offers the 42.2%, largest portion of its ODA, to South and Central Asia.
 - Other countries prioritize giving ODA to countries 'South of Sahara' followed by 'South and Central Asia'. Spain offers a surprisingly large portion of its ODA to 'Latin America and the Caribbean', which may be a remnant of its colonial-era ties.

PROJECT SUMMARY AND EVALUATION:



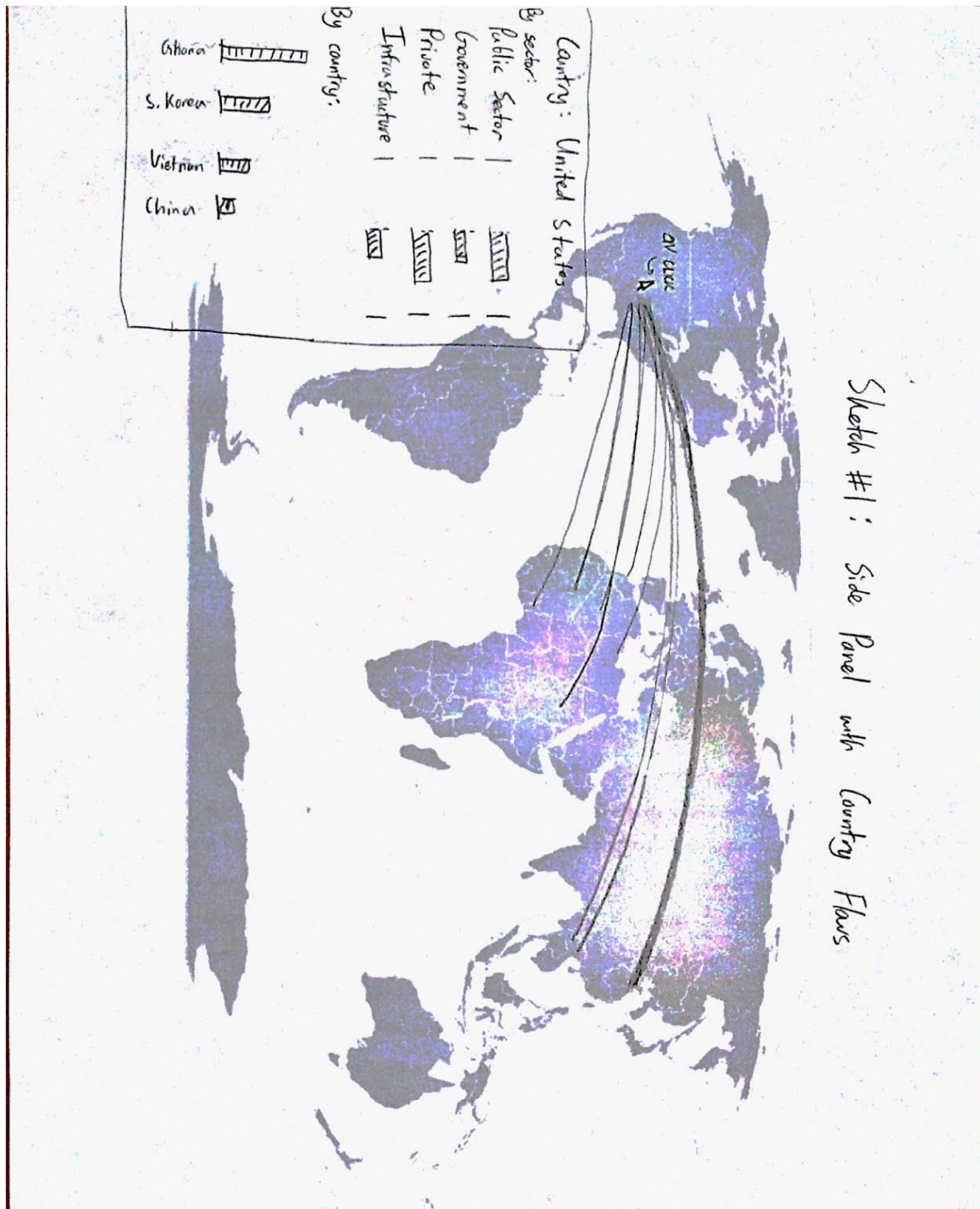
Through our second project, we produced a complex and intuitive way to visualize aid data. Our project includes **multiple, coordinated views**, by including a map view and a pie chart view of the selected country. In addition, our map view has **significant complexity**, because it supports the animation of the web of connections representing by the flow of aid data.

Of the major visualization features in class, our visualization excels in providing several ways to interact with data. **Linking & brushing** is supported by the selection of a single country: this selection creates a unique pie chart display and also a web of animated lines from the country. After a particular country is selected, moving the cursor over another country will highlight the line between the selected country and the country that is being hovered over. This highlighted status also changes the tooltip display information to be specific to that particularly link and bilateral aid flow. This dynamic tooltip allows for the production of **details-on-demand**. The separate pie chart view also provides details on demand through linking.

Finally, we implemented a **drill-down/filter capability**, such that only data within a certain value range is shown. This filter was achieved through the use of a range slider, which also dynamically redraws and animates the map with the new bounds.

APPENDIX: SKETCHES

SKETCH 1:

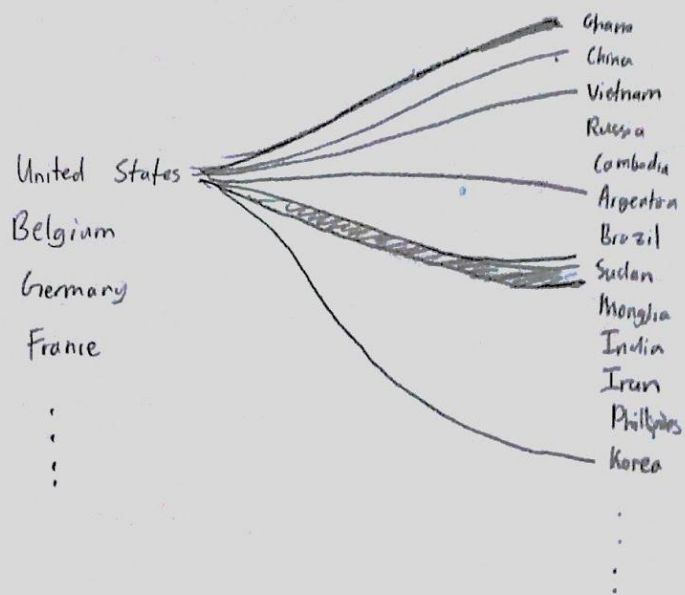


SKETCH 2:



SKETCH 3:

Sketch #3: Line Diagram



By sector:

Public

Government

Private

Infrastructure

