Juncheng Pan, Josiah Wong
CAP 6105 Final Project Read-Me

Synthesizing Sketched Diagrams for Natural Images Using Semantic Constraints

Introduction

The purpose of this file is to present steps on how the user can use the programs for our Sketch Synthesizer to collect original hand-drawn sketches, annotate natural images, and construct composite sketches that represent scenes that reflect constraints from an annotated image.

There are three programs in our system that correspond to the three phases of our work:

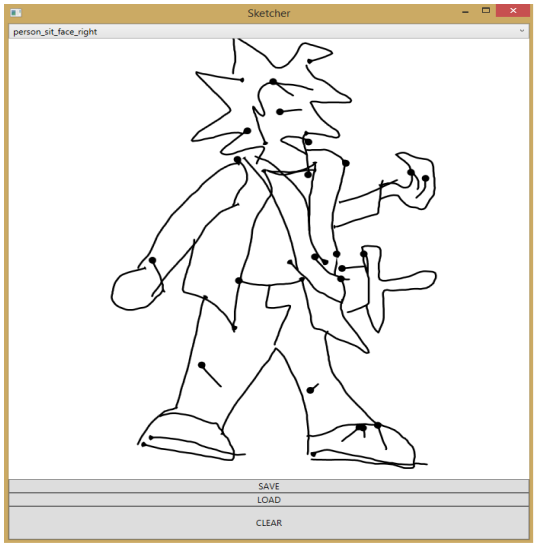
1) Sketcher – This program collects pen strokes from the user and stores these strokes as a sketch for later use.

2) Annotator – This program allows the user to derive constraints between objects in a natural image by processing annotation marks from the user.

3) Synthesizer – This program has three functions. The first is converting sketch samples from the Sketcher program into complex objects that can be manipulated. The second function is arranging complex objects into an Artificial composite sketch scene (two people chatting around a table). The third function is to arranging simple sketches to form a scene that conforms to constraints derived from image annotation marks from the Annotator.

Sketcher

The purpose of the Sketcher program is to allow the user to draw an image and save it for later use. After opening up the application, the user need only to begin drawing with a mouse, finger, or stylus in the white drawing area. The user interface looks like this (that's a rendition of a pirate – don't judge!):

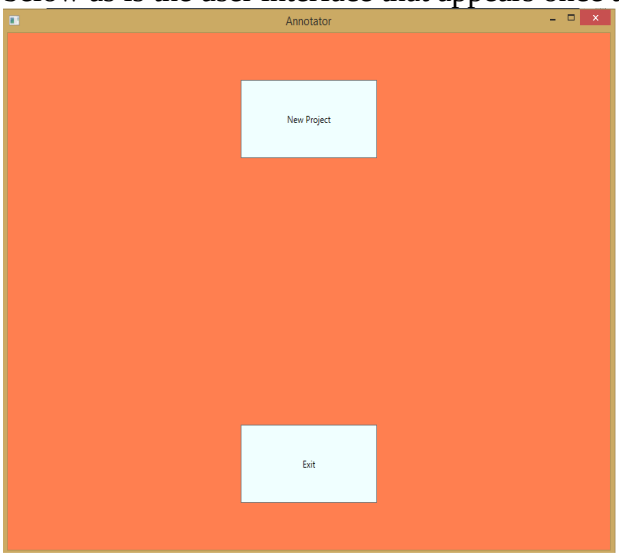


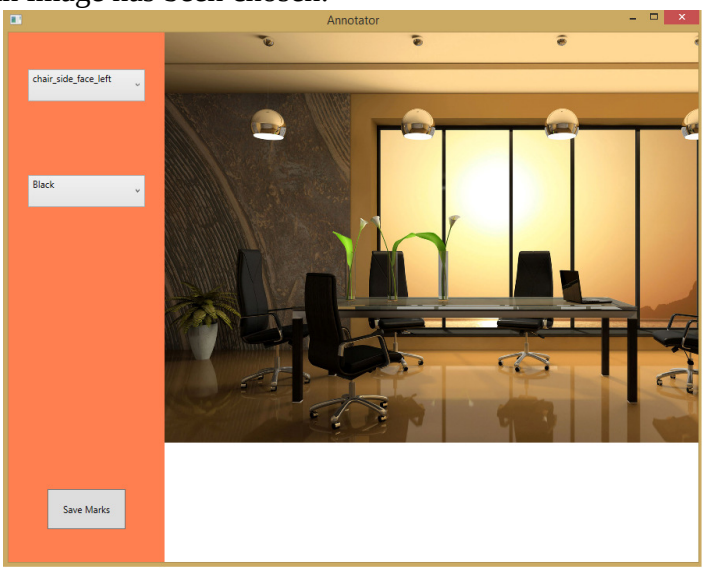
The interface is simple. Draw anywhere in the white ink canvas. The drop down menu displays a list of recognizable objects. The save button saves your image to a file called “test.data” in the directory Sketcher2/bin/Debug. This file can be renamed and moved wherever you desire. The clear button clears all strokes. Alternatively, individual strokes can be erased using a scribble erase gesture.

Annotator

The Annotator program allows the user to draw annotation marks over a natural image and save the results in order to capture constraints between objects in the image. The synthesized sketch to be created will conform to these constraints.

Once the program is run, the user is presented with two options: create a new project or exit. Clicking the Exit button closes the program. Clicking the New Project button opens a file dialog window where the user can choose an image to annotate. The user interface for the start menu is shown here below as is the user interface that appears once an image has been chosen:





In the annotation UI, the drop down menu shows which types of objects can be recognized by the application. We limited our program to office space objects as our proof of concept.

The colors drop down menu changes the color of one's future strokes in case certain colors show up better in different images.

The Save Marks button saves annotation results to a file called “scene.data” which can be renamed and moved according to the user's choice. This button should not be pressed until all annotation marks have been drawn.

In order to annotate an image, the user only needs to draw strokes over the image. (The application window can be resized to encompass the whole image for large images). The purpose of these annotation strokes is to help the computer understand where the common components of a particular object are. For example, our office space domain can recognize table objects which are annotated with three strokes: one for the table top and two for the front-most legs. Even though there are millions of table variants, every (normal) table has one table top and at least two legs.

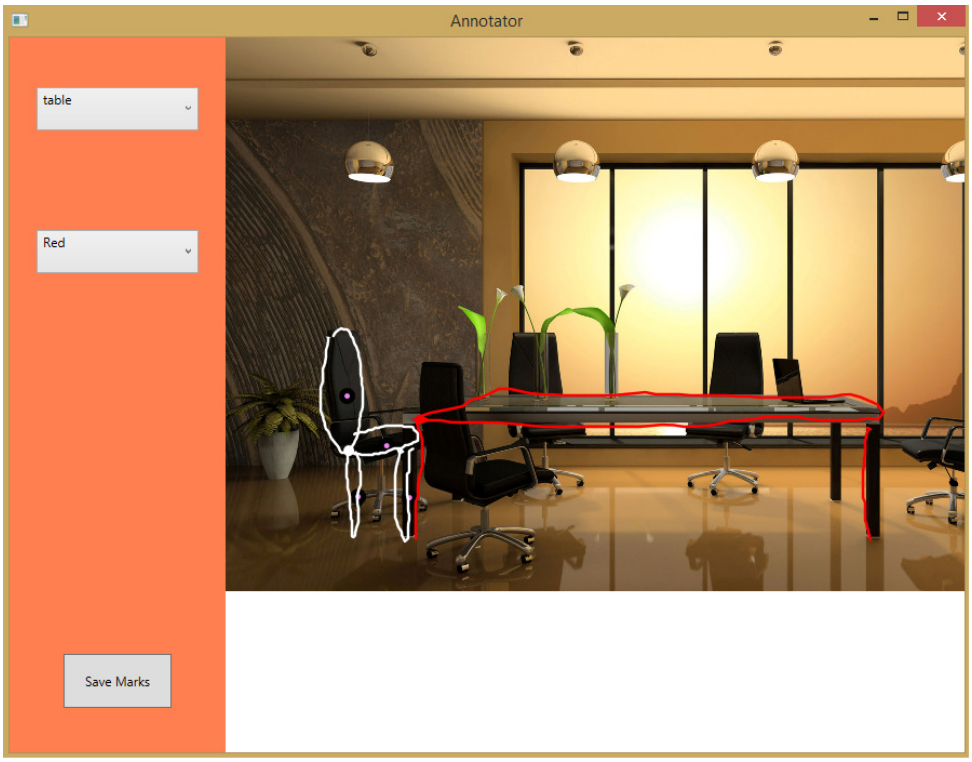
For our proof of concept office scenario, chairs are annotated with four strokes (one for the chair's back, one for the seat, and two for the legs), tables are annotated with three strokes (one for the table top and two for the legs), and a person is annotated with six strokes (one for the head, one for the back, two for the arms, two for the legs). (For a person, the annotation mark for a one arm must be to the left of the person's back and the annotation mark for the other arm must be at least partially to the right of the person's back. This is super hacky, but it gets the job done.)

The user must use one unistroke to surround one component of an object (say a chair in which case one stroke surrounds the back, one stroke surround the seat, etc.) If the user is feeling lazy, a single line can denote narrow components like a chair's legs (instead of using an ellipse to surround the leg).

Once all components of a single object have been annotated, the user can double tap on the screen to have that object recognized. If the wrong number of strokes have been used to annotate this object, the program will output a message box saying so. Otherwise, colored circle objects will appear at the centroid locations for each of the object's components.

If annotation marks must be erased, the user can erase them with a scribble erase gesture. Annotation marks for objects that have already been recognized cannot be erased.

Below is an example of one annotated chair that has been recognized and one table that is being annotated and has yet to be recognized:



Once all objects have been annotated, clicking the Save button saves annotation results to the file “scene_structure.data”.

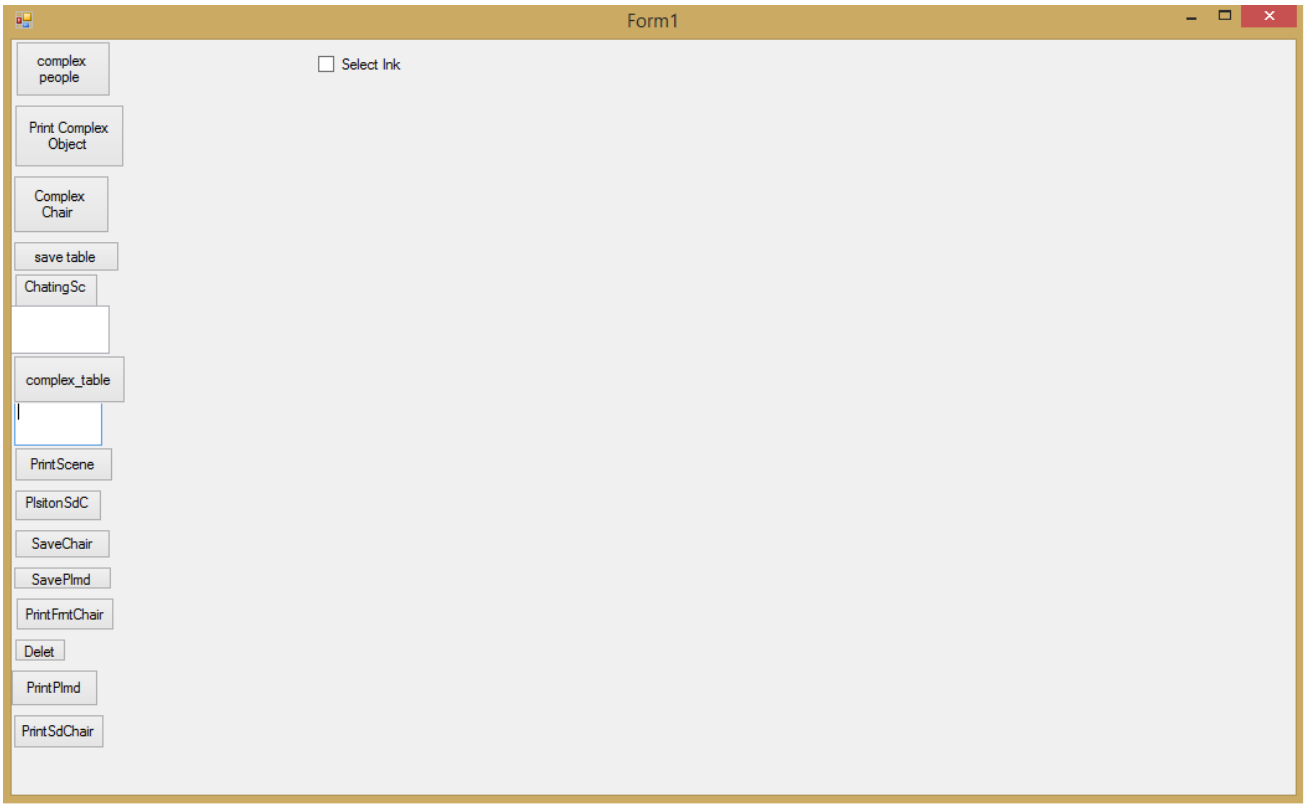
Synthesizer

The Synthesizer takes image annotation information and original complex sketch samples and compiles a composite sketch from the sketch samples that obeys the constraints from the annotation information.

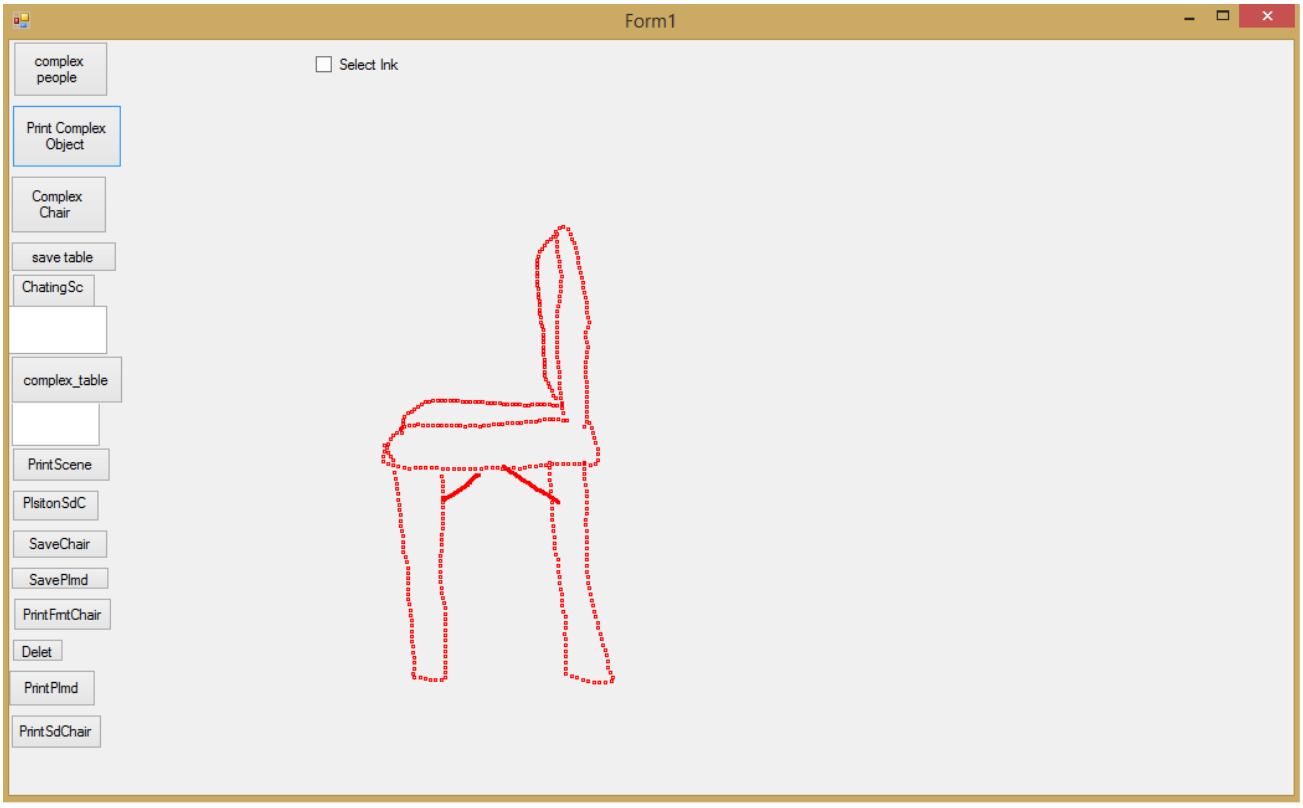
There are two main functions for the Synthesizer program: creating complex objects (to be manipulated) and synthesizing sketches.

Creating Complex Objects

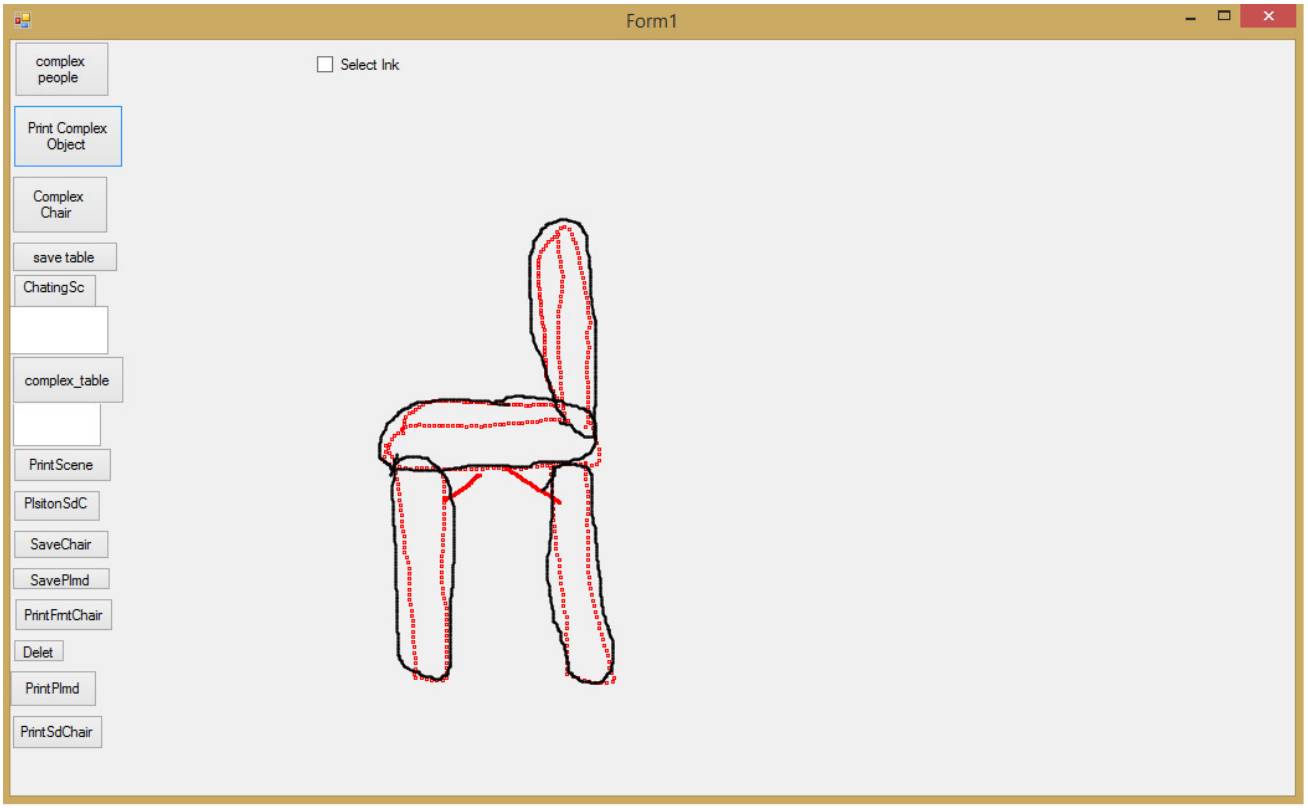
The user interface for the Synthesizer program is below:



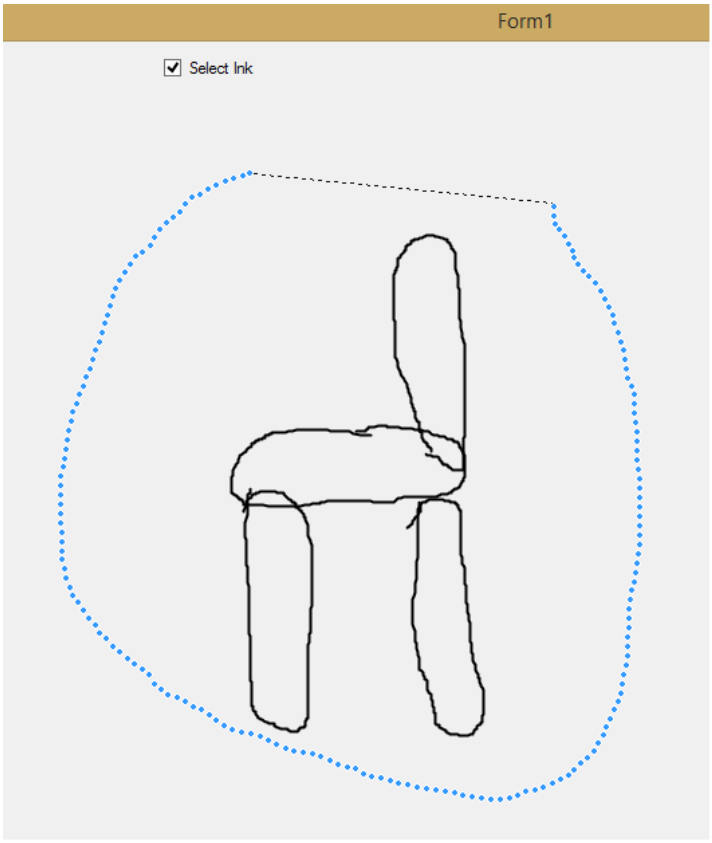
To create a complex object, we first load it by clicking “Print Complex Object” and opening a sketch sample through the open file dialog. The sketch then shows up in the manipulation area:

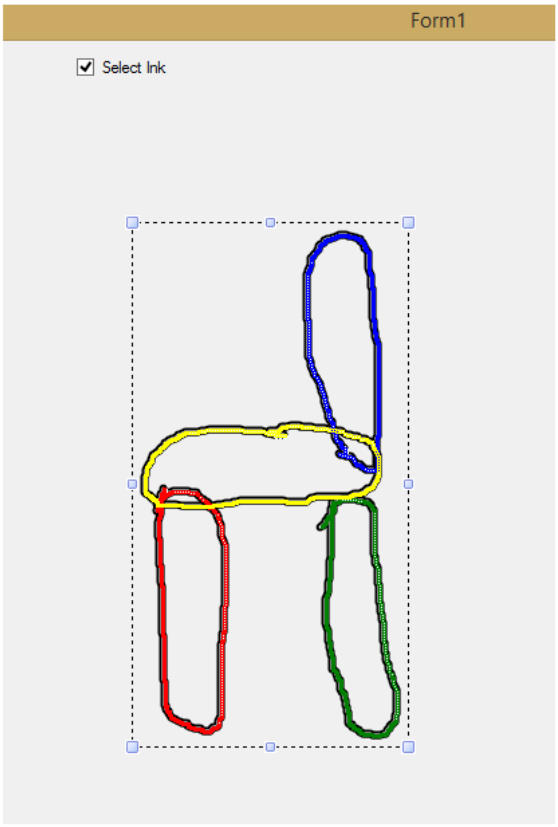


The user can then annotate this sketch using finger, mouse, or stylus input. The sketch is annotated the same way that the corresponding object in a natural image would be annotated.



Next, the Select Ink check box should be checked and using mouse, finger, or stylus input, the user must draw a circle around all annotation marks to select them.





There are three options for our office scenario: complex chair, complex person, complex table. The user can click the appropriate button and then save this complex object through the open file dialog box that pops up. After this, both the original sketch and the annotation marks appear. These marks can be erased by clicking the “Delet” button and then dragging one's input device across the strokes (the eraser tool is enabled at this point). To disable the erase button, uncheck the Select Ink check box (or if it is unchecked, check it and then uncheck it).

Synthesizing Sketch

There are two functions that provide two options for synthesizing a sketch

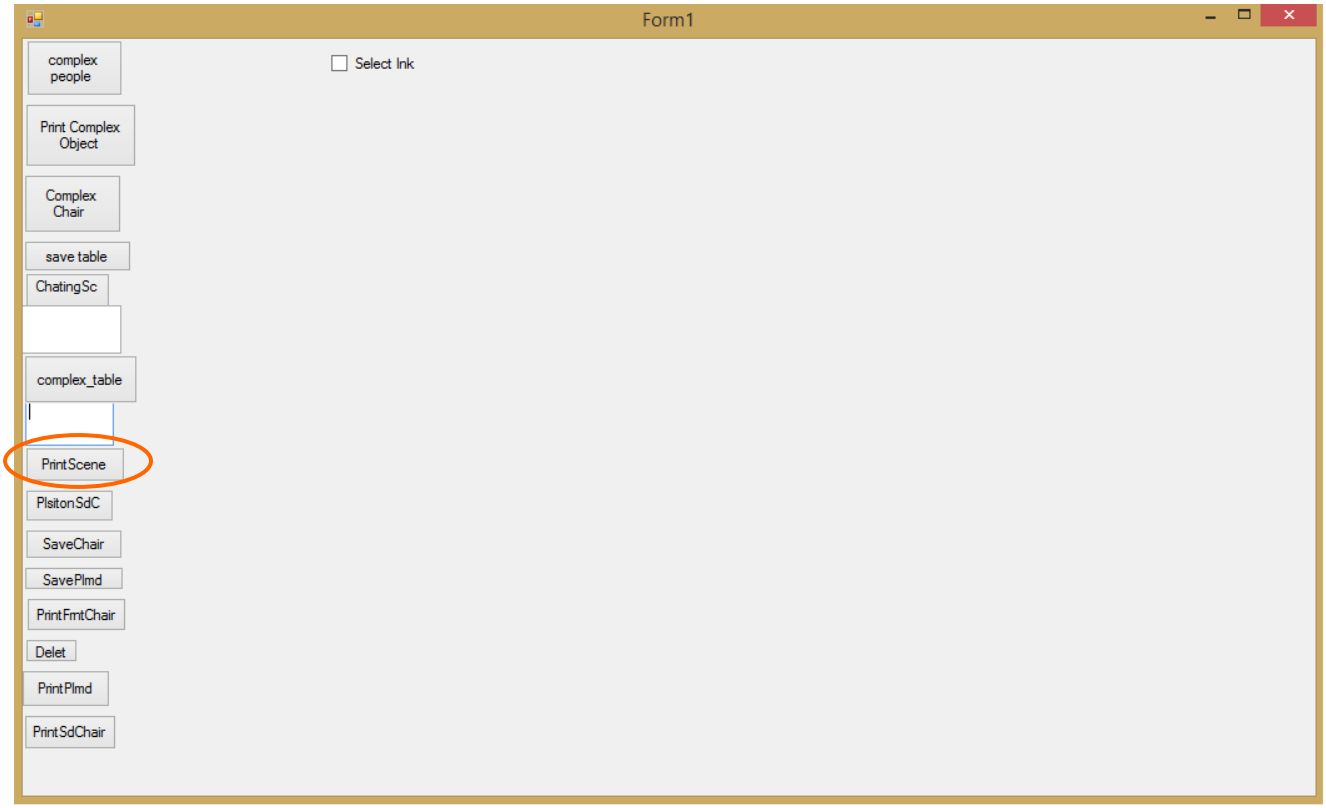
(1) Synthesize simple sketches according to an annotation of an image
The button “PrintScene” is in charge of this function. This function arranges simple sketches to form a scene that conforms to constraints derived from image annotation marks from the Annotator program.

Here are the instructions for using this function:

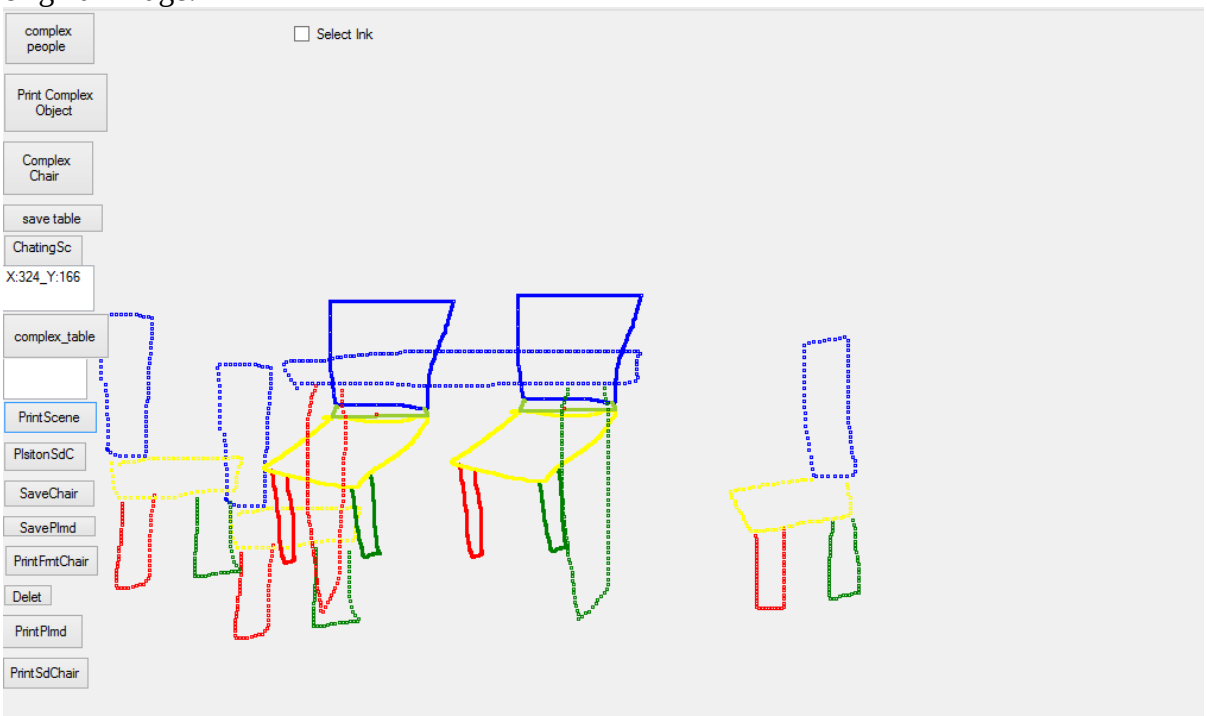
(a) Run the program under the directory :

“ Final Project - Juncheng Pan and Josiha Wong\3_Synthesizer\WindowsFormsApplication1,

And the interface will be like this:



Suppose we have annotated an image, and we save the annotation outcome as “scene_structure.data”. Please click the “PrintScene” button, then program will open a dialog to allow you to select an image annotation you have annotated, for example “scene_structure.data”. After that, the program will generate a synthesized scene that consists of simple sketches that reflect the structure information in the original image.



(2) Synthesize complex sketches to form an artificial scene

In theory we can support synthesizing complex sketches to form any scene, however, the because of limited time we only synthesized complex sketches to form an artificial scene, which is “two people sit around the table chatting with each other”.

This function is very fun; you can choose any complex sketch object to form the scene.

In order to run the function, firstly you need to create five complex object sketches as we have mentioned before. You need to create two complex people, two complex table, and two complex table, and store the output (the complex object data structures), to a specific directory.

You need to save two complex people sketches(right view people, and left view people) as “Left_people_model.data” and “Right_people_model.data”, respectively, in the directory “\Debug\complex_human_model”.

Note that the Debug folder is in the following directory:

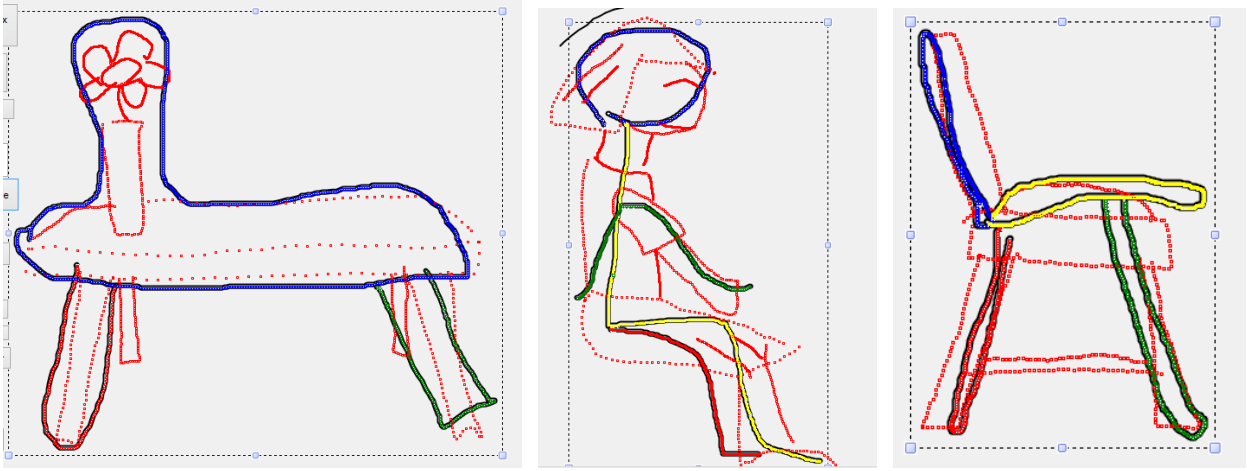
\Final Project - Juncheng Pan and Josiha Wong\3_Synthesizer\WindowsFormsApplication1\ WindowsFormsApplication1\bin\Debug

Similarly, you need to save two complex chair sketches model in “\Debug\complex_sideviewChair” as “Left_chair1.data” and “Right_chair1.data” respectively.

And you need to save the complex table sketch in “\Debug\complex_table” as “Table1.data”.

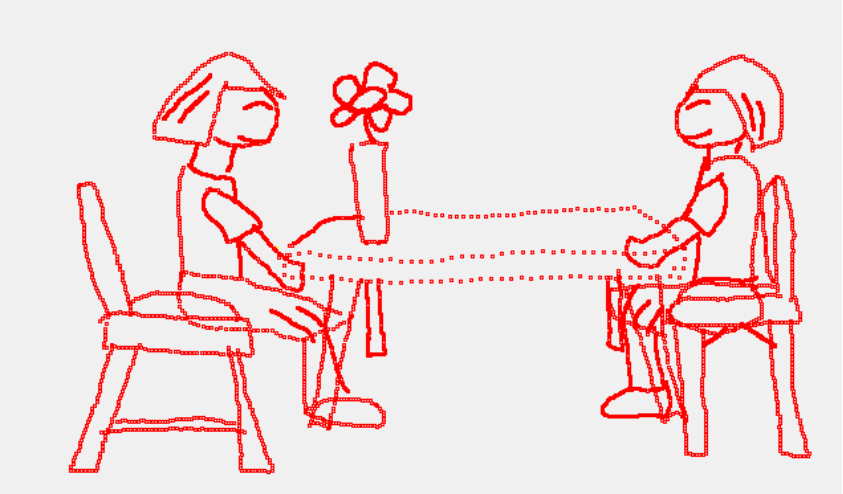
After that, you click the button “ChatingSc” button. Then, the program will show the synthesized scene that consists of your complex objects.

Below is the example(note that we have stored the complex sketches model in the specific directory, so you can directly click the “ChatingSc” the button to show the scene, but you can also customize your sketch using the steps mentioned above) ” .



Create complex sketches and stored them in specific directory

The figure below shows the synthesized graph based on the complex sketches objects you have created.



Synthesized scene consists of the complex sketches objects