# Literature Review 3

## Primary paper:

### DrawFromDrawings: 2D Drawing Assistance via Stroke Interpolation with a Sketch Database

BibTex:

```
@article{article,
author = {Matsui, Yusuke and Shiratori, Takaaki and Aizawa, Kiyoharu},
year = {2016},
month = {04},
pages = {1-1},
title = {DrawFromDrawings: 2D Drawing Assistance with a Sketch Database},
volume = {23},
booktitle = {IEEE Transactions on Visualization and Computer Graphics}
}
```

## Secondary paper:

### Real-time Drawing Assistance through Crowdsourcing

```
@article{Limpaecher:2013:RDA:2461912.2462016,
  author = {Limpaecher, Alex and Feltman, Nicolas and Treuille, Adrien and Cohen,
Michael},
  title = {Real-time Drawing Assistance Through Crowdsourcing},
  journal = {ACM Trans. Graph.},
  issue_date = {July 2013},
  volume = {32},
  number = {4},
  month = jul,
  year = {2013},
  issn = {0730-0301},
  pages = {54:1--54:8},
  articleno = {54},
  numpages = {8},
  url = {http://doi.acm.org/10.1145/2461912.2462016},
  doi = {10.1145/2461912.2462016},
  acmid = {2462016},
  publisher = {ACM},
  address = {New York, NY, USA},
  keywords = {crowdsourcing, interactive drawings},
}
```

**Summary:**

Limpaecher et al. [Limpaecher et al. 2013] have presented a unique crowdsourcing approach of using social game mechanics to grow and use a dataset of drawings, which is an iPhone game *DrawAFriend*. In the game, players at first choose to start with either a Facebook friend or an anonymous stranger. Then the player is given four pictures which she/he can draw. After this, the player can trace the image. At any point, the player can hide the photo and see the drawing on its own. Once finished, the player sends the drawing to the friend or anonymous player with whom is playing. The friend or anonymous player is prompted to guess the identity of the other player's drawing. Limpaecher et al. hypothesize that a good drawing is much more likely to be guessed correctly than a bad drawing. They introduced a method to extract stroke-level artistic consensus from a large drawing corpus and develop a stroke correction strategy with two phases: Consensus Finding and Interactive Correction. The method will analyze the drawing corpus to compute a correction vector field that for any location, points towards a nearby consensus of strokes so that develop a real-time self-correction touch interface. Furthermore, in order to solve the "fat finger" problem that occurs with a touch device, they introduced a simple method to correct strokes based on the nearby consensus while maintaining the stylistic choices of the artist. Especially, auto-corrections run interactively and appear nearly invisible to the user while seamlessly preserving artistic intent so that the resulting strokes feel more like the intent of the user than the raw original strokes. When they found that some strokes are purely stylistic in nature, they do not want such strokes to have parts pulled towards the consensus so that they just leave it alone. In the end, Limpaecher et al. evaluated our stroke correction algorithm through crowd sourcing, by observing how it influenced drawings "in the wild" by instrumenting the game itself. In general, they believe that DrawAFriend presents an unprecedented platform to perform quantitative drawing analysis at the Internet scale. However, there are several drawbacks. Until now, this game is focus on face portraits. Therefore, the non-portrait drawings are still filtered out by hand currently. It is still essential to find several avenues to improve and generalize our correction vector field model.

Matsui et al. [Matsui et al. 2016] have designed an interactive assistive interface for 2D drawing, called *DrawFromDrawings*. This interface can provide users with visual feedback for assistance in 2D drawing using a database of sketch images. There are two kinds of feedbacks that the deformation feedback, which allows users to replace their strokes with interpolated strokes, and the suggestive feedback, which overlays interpolated strokes onto users' strokes. In the DrawFromDrawings system, the user repeats three steps: drawing and retrieval that the user start to draw in user canvas and select a favorite sketch displayed in the reference canvas, exploration that the user can invoke the assistance function with a reference sketch, and refinement that the user can utilize interpolated strokes via either deformation or suggestive feedback, until satisfactory convergence occurs. According to the regions of interest (ROIs) in the user and reference sketches, DrawFromDrawings automatically detects as-long-as-possible (ALAP) stroke segments and the correspondences between user and reference sketches that are the key to computing seamless interpolations. The stroke-level interpolations are parametrized with the original user strokes, reference strokes, and potential new strokes that are the novel strokes created by warping the reference strokes based on the shapes of the user and reference ROIs. Matsui et al. resample the corresponding user and reference ALAP segments and compute an interpolated position q for each resampling point as:

$$\mathrm{q} = (1 - \beta)p^u + \beta((1 - \alpha)W_r^u(p^r) + \alpha T_r^u(p^r))$$

where $W_r^u$ is a warping function, $T_r^u$ is a similarity transformation, $p^u$ and $p^r$ are corresponding resampling points in the user and reference ALAP segments, respectively. α is a parameter indicating the degree to which the outline shape of resulting strokes resembles that of the user strokes and β is a parameter indicating the degree of preservation of the user strokes. Matsui et al. [Matsui et al. 2016] performed three user studies, investigating drawing improvements with DrawFromDrawings, comparing between the deformation and suggestive feedback modes, and analyzing utility of the stroke interpolation. From the first two user studies, the tendency is that the suggestive feedback was used to develop ideas and maintain stroke style while the deformation feedback tended to be used for two cases: one case is to quickly finish regions with complicated texture such as fur of animals in the task object drawing study, the other case is that inexperienced participants often used the deformation feedback to complete entire drawing. From the third user studies, α varied independently of the shape differences, β plays an important role for sketches with complicated texture such as fur. However, the stroke correspondence detection and interpolation has a limitation. Because this method relies on the convex hulls of ROIs, an association failure might occur, such as when the user-specified ROIs are thin and concave with significantly different orientations. And the sketch image retrieval also has several limitations. Therefore, drawing assistance without references such as considering temporal order of drawing might be an interesting direction for further drawing skill development.

**Relationship:**

Matsui et al. cited Limpaecher et al.'s paper that the reason why Matsui et al. want to investigate whether or not the deformation of user strokes helps with drawing is that it is possible aggressive deformations will overwhelm a user's own strokes at the expense of creativity. Because Matsui et al. is different from Limpaecher et al.'s corrective feedback, which is small, possible unnoticeable, deformations to user strokes for beautification purposes. On the other hand, Matsui et al. observed that the suggestive feedback was used to develop ideas and maintain stroke style. This observation was the same in Limpaecher et al.'s paper. In the part of limitation, Matsui et al. shows that possible assistance from the system is still limited to the sketches available in the database. This also is a limitation for Limpaecher et al.. However, there are several difference between Matsui et al. and Limpaecher et al.. Firstly, Matsui et al. want to be able to refer to rasterized sketch images where much of meta stroke information such as stroke order is lost while Limpaecher et al.'s tracing paradigm results in a set of pre-aligned drawings that collect individual strokes represented as polylines along with timing information. Secondly, for *DrawFromDrawings*, this system allows users to either replace their strokes with interpolated strokes or overlays interpolated strokes onto their strokes. But for *DrawAFriend*, this system requires no new user interaction paradigms that it appears "invisible" to the user and will automatically correct the user's strokes.