Enhanced Auto Completion of Hand Drawn Sketches

[1] Mohammad Zulqurnain, [2] Dr. Imran Amin

[1,2]Shaheed Zulfiqar Ali Bhutto Institute of Science and Technology, Karachi Pakistan

[1]mohammad.zulqurnain@gmail.com
[2]amin.imran@gmail.com

Abstract— Sketching is one of the most effective way to communicate art and imagination of an individual. It adds a sense of realism to the object in this work our system helps the user drawing repetitive structures with in a sketch like hatching, stippling, cross blending, broad stroke, and object contours which saves time on the part of artist and he can concentrate on the non-repetitive sections of the sketch. Our system gracefully fit the probable pattern repetitions to the changing and non-repetitive context in the drawing and lessens the number of manual strokes that has to be performed by the artist while maintaining the aesthetic aspect of the drawing.

Keywords—Auto completion, hand drawn sketch, digital art, correlation, Synthesis, Workflow, User interaction

I. INTRODUCTION

Visual art is the most expressive medium of communicating one's creativity in today's world whether it be infographics, eBooks, web animations or games and with a fast evolving technology and with regard to modern era assistance and refinement in the field of arts and graphics, is felt more acutely due to the increase in the pace of workflow required. The process of sketching involves drawing certain patterns like hatching, stippling and cross blending which are necessary to add a sense of liveliness in the object but they incur a significant portion of time on the part of artist in the past decade several efforts have been made to make life easy for art community via automated interactive drawing tools [1] which can assist them in auto completion of drawing of repeated structures for instance stipples, hatches, or brushes [2] with in single drawing so that he can channelize his creativity to the better half of an aesthetic spectrum by successive workflow [3] and being able to express himself in a more lucid manner with the help of spatial and temporal hints provided by the system.

The goal of this work is to enhance the previous auto completion technique [2] so that it can cater to the abrupt changes in context surrounding the stroke in the drawing in a real-time fluid manner for the strokes provided by the user unlike [4] in which batch sequencing was used in which the user has to wait while the batch to be and later that was mapped to desired region. In our system, user can draw and can get real-time predictions while he is drawing which can be seen in the fish exemplar in figure 1 user's subsequent strokes get recorded dynamically and are used for predicting the future strokes with in dynamically changing context adjustment and the user can approve whether he wants to implement prediction or keep drawing.

Our system predictions are based upon the spatial (contexts and proximity), temporal (closest operation in the history) and appearance parameters (color, direction, distance, pressure and location) [5] following predictions are adjusted to the abruptly changing constraints [6]. In this work, surrounding contexts are iteratively adapted to for better fit. Succinctly this work contributes towards:

- An enhanced auto completion system to predict artist's consecutive strokes
- Adjust the prediction to the abruptly changing context

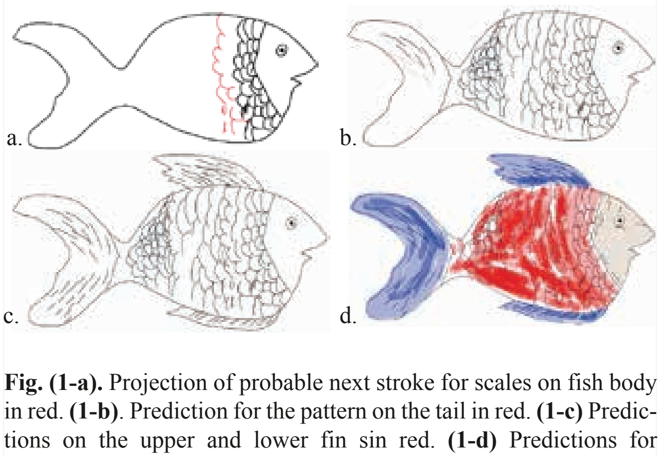


Fig. (1-a). Projection of probable next stroke for scales on fish body in red. **(1-b)**. Prediction for the pattern on the tail in red. **(1-c)** Predictions on the upper and lower fin sin red. **(1-d)** Predictions for coloring in thick red strokes on fish body.

Following is how the sections are divided in this work in section II we discuss the previous work, in section III we discuss our user interface, in section IV we discuss our method overview, in section V we discuss the basic technique behind our work that is workflow texture synthesis, in section VI we discuss the results, in section VII we discuss about discussion, section VIII contains conclusion, section IX contains future work and section X contains the acknowledgments.

II. PREVIOUS WORK

A. Automation of patterns and textures

In the work [1], author has devised a mechanism to mark a group of objects with their respective movement (which has both global and granular motion aspect) in order to cater them in the dynamic texture in real time.

B. Data analysis

In the system [7], author has devised a mechanism to project prediction under the user sketch based on the edges similarity with the database of 30000 most common objects in everyday life images which are based on closest edge match. Both global and local context of the drawing is compared to the image and in doing so weights are applied for the optimal predictions based on the curvature of the edges and their relative structure to each other.

C. Local similarity and Global similarity

Author has implemented the system for real-time objects simulations in live scenarios in generic natural instances like piles, bushes, particles and etc. [6]. It deal with exploiting high level structures through spatial-temporal paradigms for output constraints plus low level details through small input operations for instance motion and arrangement of elements. Furthermore, it aims to enhance repetition in efficient and dominating manner via constraint optimization techniques as well as synthesis algorithm for satisfying user assigned controlling.

D. Auto completion in drawing

In the work [2], author has developed a system to keep track of users respective strokes at dynamically at runtime and eventually use them to predict next stroke, on the basis of temporal, spatial and appearance similarity however it has a limitation when it comes to dynamically changing surrounding context in the drawing.

E. Workflow Analysis

In the work [8], author has worked on a system which can retain the previous state of the system by taking back image of the scenario at regular intervals it basically consists of two aspects one is artifact which stores the particular state of the work element at unique points and the other is context which stores the application state at global level. Causality caters to different discrete relationships like transitive between users input and its value and their respective outcomes. Moreover, this work is rigid enough to cater to the paradoxes that may arise in the work flow in case of parent child relationship when respective parent state is deleted and it solves it by using parallel histories.

III. USER INTERFACE

We have come up with an auto completion system that learns the repetitive patterns drawn by the artist as can be seen from figure 2 our auto completion system has a simple interface which involves causality controls like undo(a)/redo(b) for navigating pass the drawing flow, basic drawing application tools like drawing color(h), drawing tool(i)(m)(n) parameters options and auto completion tools which includes display label for predictions settings for prediction operations(d), to yes / no(e), setting number of history operations to consider(f), setting spatial window for the algorithm(j), setting appearance parameters effect(k) and setting temporal parameters effect(l).

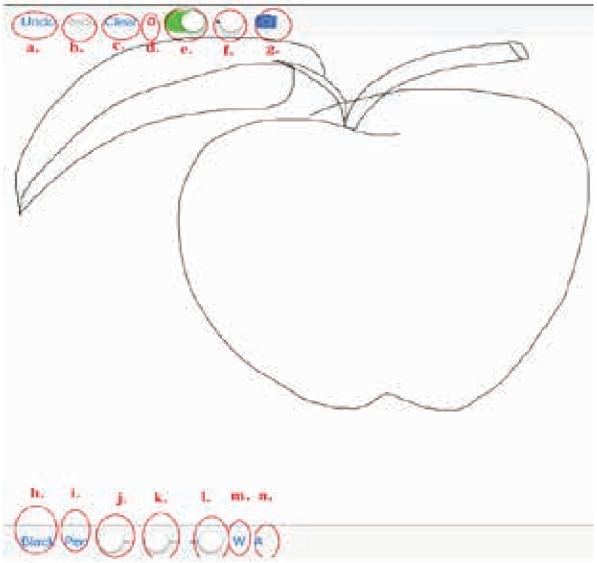


Fig. (2-a). Undo **(2-b)**. Redo **(2-c)**. Clear **(2-d)**. Predicted operations **(2-e)**. Auto completion toggle switch **(2-f)**. Temporal window **(2-g)**. Screenshot **(2-h)**. Color select **(2-i)**. Draw tools **(2-j)**. Spatial window **(2-k)**. Appearance parameter weight **(2-l)**. Temporal parameter weight **(2-m)**. Stroke width **(2-n)**. Alpha setting.

A. Auto completion

As it can be seen from figure 3 to figure 7, as the user draws strokes in sketching the prediction appears in red color over the canvas at the appropriate location and they become more accurate as the user keep on drawing the sketch. While the user is drawing the predictions will keep on appearing in real-time and if he wishes to approve them he performs double tap gesture over them as can be seen in figure 4 to figure 5. While drawing the predictions confines themselves in nearest context strokes i.e. contexts can be global (largest stroke operation) for instance leaf stem in figure 10 (b) & (c) or local for instance window frames in 8 (c).

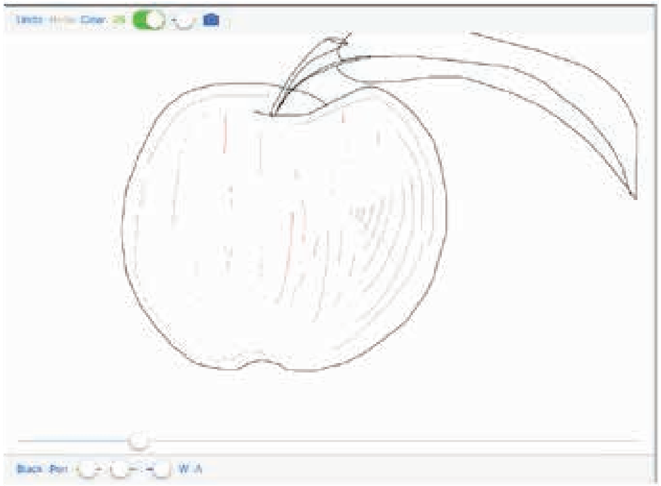


Fig. (3). Initial sketching on apple with predicted strokes in red color.

First, we draw the outline as in figure 2 then we start shading it as can be seen in this figure the predictions starts to appear after 30 strokes since we have adjusted the temporal window to 30 operations. User can accept predictions via double tap gesture on their device.

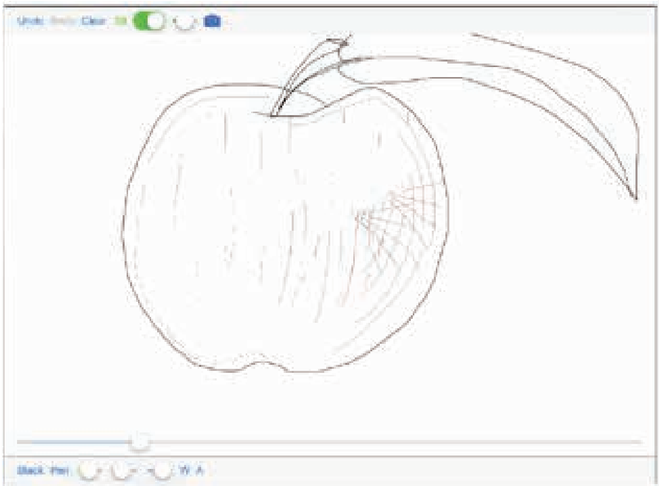


Fig. (4). Complex hatching strokes being predicted in red color and displayed in red color with respect to nearest context strokes.

Hatches are predicted with respect to the nearby normal context operations.

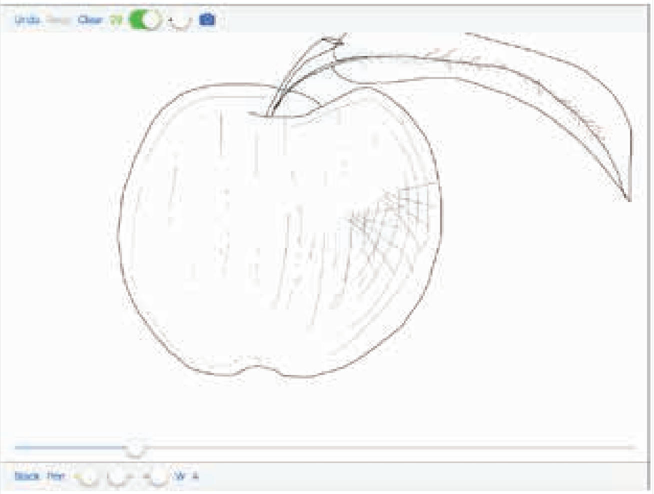


Fig. (5). Stippling with global context stroke with predicted strokes in red color as reference.

It can be noticed that the stipples inside the leaves are following their nearest maximum strokes (leaf middle line) as well as nearest contexts more tightly when predictions are made as we have increased the spatial window due to which is considering multiple nearest context strokes.

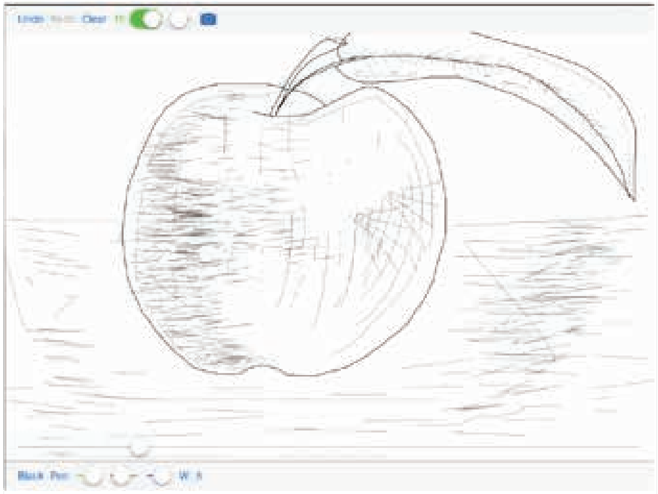


Fig. (6). Temporal window is increased from 30 to 40 prediction operations in red color.

As we increased the temporal window there is an increase in the number of predictions as can be noticed.

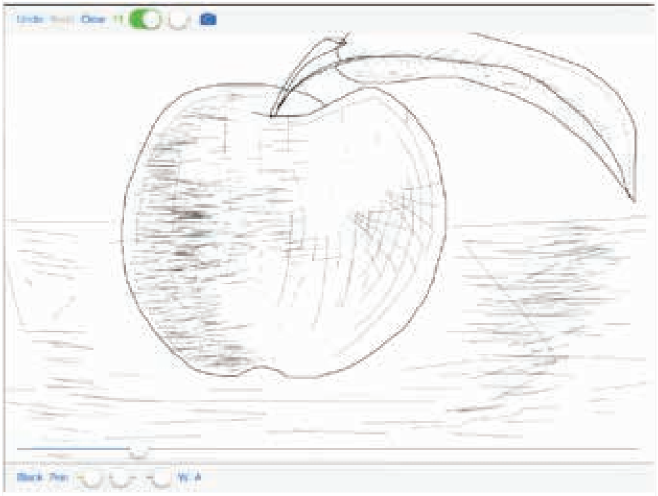


Fig. (7). Finalized apple sketch

In total there has been 2000 strokes out of which 900 are predicted strokes.

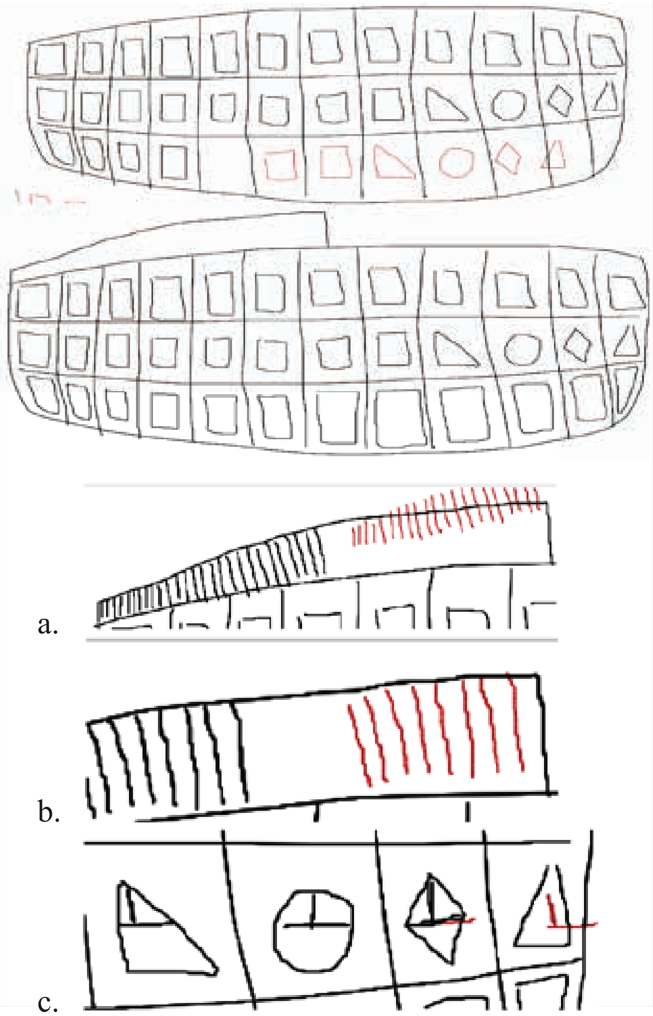


Fig. (8-a). With the increase in the number of strokes the predictions in red adapts to the surrounding context strokes (nearest strokes greater than 2x the size of current operation). (**8-b**). The prediction in red color fits the surrounding context **(8-c)**. The prediction fits the non-homogeneous surrounding context as can be seen from the non-uniform surrounding contexts in the drawing.

IV. METHOD OVERVIEW

The theme of our work is based on looking for repetition in drawing similar to texture synthesis [6-5], [9]. As discussed in previous work section utilizes geometric relationships within data (i.e. set of stroke operations in our case) and in doing so considers both global aspect (relationship with largest structures with in the drawing) and local aspect (relationship among the repetitive structures themselves). In our system each drawing operation (i.e. stroke performed with continuous touch) is a set of points which we call sample. Each time the user sketches a stroke on to the it is saved in the history of operation which has a spatial window for recent operations to consider based on their dissimilarity with the current operation and the output is the predicted operation owing to the nearest surrounding context operation figure 9 both global and local [10]. The main feature of our work is that it considers the non-homogeneous contexts in the surrounding neighborhood as seen in figure 8.

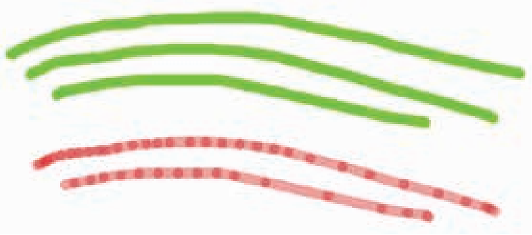


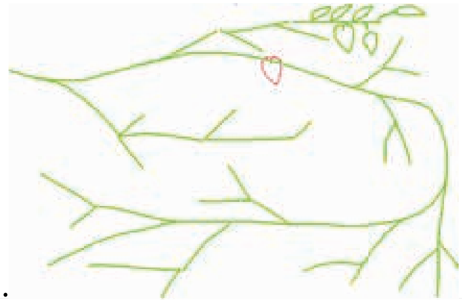
Fig. (9). Predictions (red transparent lines) are made on the runtime as the user draws the strokes (green lines) on the canvas.

V. WORKFLOW OF TEXTURE SYNTHESIS

Two aspects of our work are as follows:

A. Locality – It refer to the nearest neighborhood of the operation (as can be seen in figure 10 (c)).

B. Context – It refers to the nearest neighborhood samples that are more than twice the size of the current operation figure 10 (a) and (b).



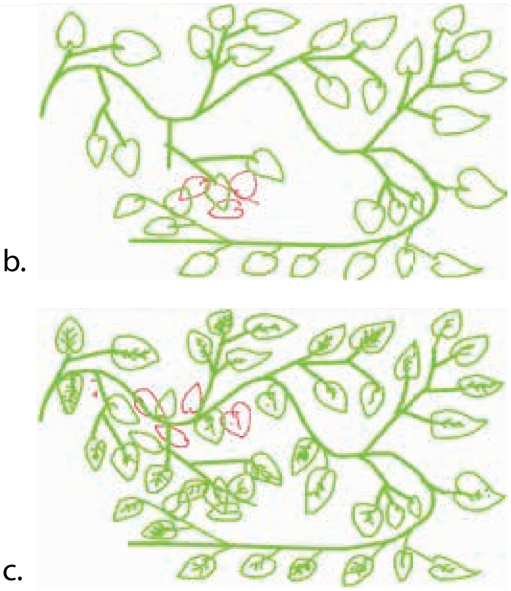


Fig. (10-a). Multiple branches sprouts out of the main stem which represents our context strokes owing to which the prediction is made in red **(10-b)**. Prediction is made in red in continuity with the context strokes **(10-c)**. Prediction is made considering both the context strokes and the relative distance from the surrounding operations.

C. Representation

History:

It is the data structure to keep track of the operations performed by the user and is represented by

h = {op} (1)

Operation:

Each operation consists of an array of sample elements (points in drawing), operation ID, and central sample and can be saved in the history array for predictions. And it is denoted by

op = {s} (2) [2]

Sample:

It is the basic element of the drawing stroke and is composed of spatial parameter (location, direction) represented as p, appearance parameter (color, size, pressure stamp) represented as a and Temporal parameter (Time, Sample ID (keeps track of sample's relative position within the drawing canvas) and Sample Local ID (keeps track of sample's relative position within the operation)) represented as t and it is denoted by s

where s E op and

u(s) = (p(s), a(s), t(s)) (3)

represents the attributes of a sample [2].

Constraints:

It is the data structure which keeps track of the coherence between the current operation geometric relation with the surrounding contexts in the drawing and is composed of sample elements offset (distance from the current stroke to nearest context stroke) and Sample Elements Local Id Pairs(-to keep track of the correct surrounding context for the current operation). It can be observed in figure 12(a), (b) and figure 13 that the constraints are represented by the red lines where there length is constraint represents offsets between the locked samples of two different operations (i.e. context operation and predicted operation). This is used to fit the subsequent prediction with their respective nearest context (figure 8 (c)) so that predictions adapts well to their non-homogeneous surrounding context operations. And is denoted by c where:

c(op)=

$\bigcup_{scurrent \in opcurrent, scontext \in opcontext} Lid(scurrent, scontext)$ (4)

Constraints:

Constraints are basically data structures in our system which keeps track of the locked order pairs between context and prediction operations and referenced by their respective sample's local IDs.

Soft Constraints:

This is the measure of percentage similarity of the Constraints offsets in current constraints between context and offset applied prediction compared to prediction operation offsets with context operation before applying offsets and is used to choose present context that is context of the prediction after it is offset applied prediction and is denoted by sc

where sc E R as denoted

sc = {(0.5,1)|c/c'} (5)

and where is c' are initial constraints related to previous context and c are the constraints related to the present context. They serve in speeding up the results with good fits.

Neighborhood:

It is the locality of the surrounding samples of current stroke and can be adjusted by the spatial window parameter (higher value will signify larger neighborhood influence). It is denoted by n where

$n(op) = \bigcup_{s \in op} n(s)$ (6)[2]

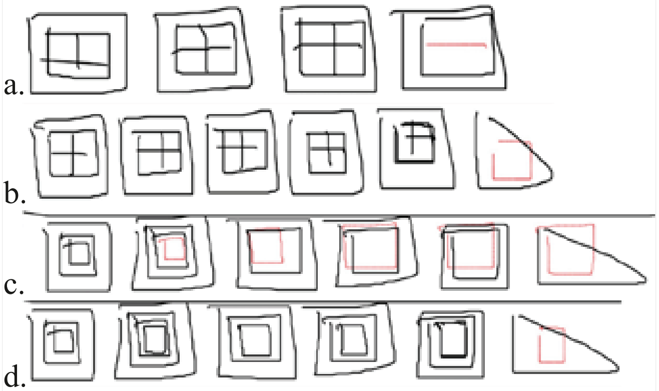


Fig. (11). Cases involving local context.

It can be seen in figure 11 that the prediction is adjusted to the local context with in repetitive context. Prediction can fit onto abruptly changing contexts as well as shown in the figure. With the passage of time the contexts prediction proceeds towards better fit. Prediction can be transformed eventually to obtain a better it with regard to the local context.

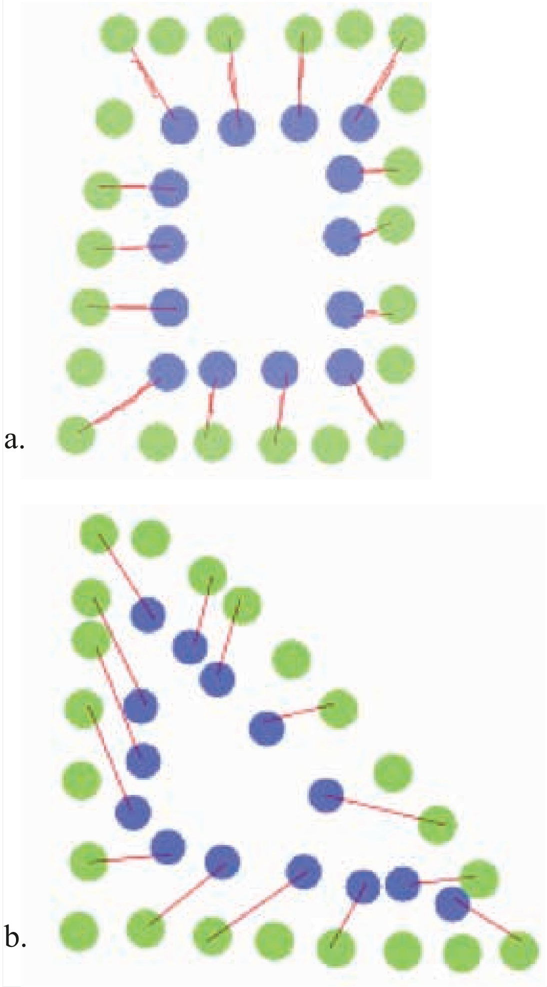


Fig. (12). Shows how the local context adjustment works in our system where green dots represent the local context stroke, blue dots represent the actual prediction and red lines represent the distance constraints which are locked before the offset is applied to the detected prediction (a). Shows how the predicted operations are locked with the surrounding context before getting offset (b). Shows how the offseted prediction with it's given local contexts adjusts with constraints data structure that we have incorporated.

Context operations:

They are the nearest operations to the current stroke which are at least 1.25 times bigger in size than the current operations are calculated at runtime and is represented by co $\in$ h where

$$co = \{(op \in h, co \in h)\ |co > 1.5\ op\} \tag{7}$$

(for initial constraints of the prediction) and

$$c(op) = \cup_{s \in op} n(s) \tag{8}$$

(for present constraints of the prediction)

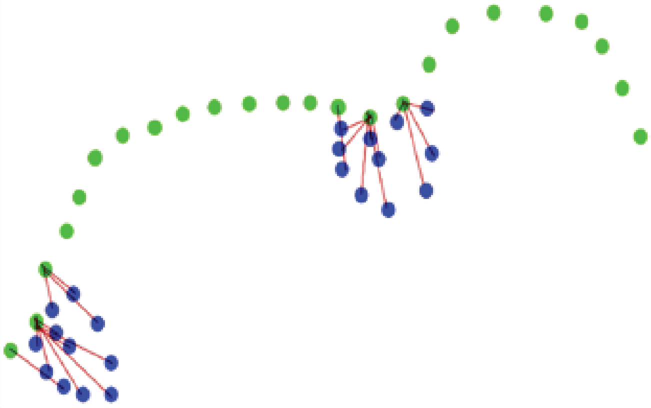


Fig. (13). Global context prediction mapping is demonstrated here where green dots represent the context stroke, blue dots represent the prediction operations and red line represents the distance offset between the predicted operations and the context operation. The operation to the left is prediction before the offset is applied to it and to the right is the operation after the offsets have been applied to it.

i. Similarity measure

Neighborhood analysis is analogous to the [6]

Sample similarity:

Similarity assessment is based on the following equation:

$$u(s', s) = (p(s', s), \partial a(s', s), ßt(s', s)) \tag{9}$$

y adjusting the values of ∂ and ß which lie in range [0,1] we can vary the effect of the appearance and temporal aspects of the drawing figure 2 (k) and (l) [2].

Operation similarity:

Since operations are the collection of the samples the similarity between the operations can be equated as:

$$u(op', op) = \{u(s', s) | s' \in op', s \in op\} \qquad (10)[2]$$

where op' is the operation from the history op'∈h before op and like s' is it's respective sample and op is current operation

Neighborhood similarity:

It relies on two aspects:

Measuring the closeness of the operation with regard to spatial window represented as:

$$|u(op_o, s_o) - u(op_i, s_i)|^2$$

and secondly aggregate closeness of the samples with in operations for nearest sample ids as :

$$\sum_{op' \in n(opo), opi' \in n(opi)} |u(opo, opo) - u(opi, opi)|^2$$

Complete equation for neighborhood similarity is:

$$|n(op_o) - n(op_i)|^2 = |u(op_o, s_o) - u(op_i, s_i)|^2 + \sum_{op' \in n(opo), opi' \in n(opi)} |u(opo, opo) - u(opi, opi)|^2 \qquad (11)[2]$$

ii. Algorithm

Input stage:

User draws the stroke on the screen by performing touch began and move gesture on the screen which conveys the respective points to the system.

Recording stage:

The system records the subsequent strokes in the history data structure for later use.

Analysis stage:

If there are sufficient samples available then proceed to the next step.

Searching stage:

Search in the history with regard to the spatial window defined then search for the least dissimilar operation to current operation in the history looking back the number of steps defined in temporal window and keep track of the match found.

Positioning stage:

Compute the distance from the predicted operation and record it as offset then add that offset to the matching operation and all the operation after it leading to the current operation.

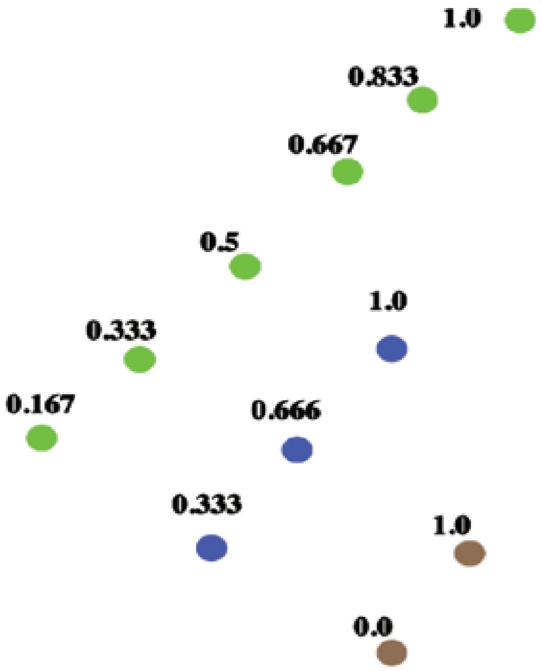


Fig. (14). Operation with samples of respective local ids are shown in the figure. Sample's local id is the measure of its relative location to other samples with in the operation.

Context operation search stage:

Before applying the offset to the prediction the order pair of samples with minimum distance in between them is searched for between predicted operation and surrounding context operation if it is within spatial vicinity, if so then that particular predicted operation within which that sample lies it's offset with the context operation is recorded for later use. After applying the offset to the predicted operation again the ideal sample order pair is searched for which has the minimum distance in between and the offset recorded before offsetting the predicted operation is applied to the prediction relative to the nearest context operation which adjusts the sample with the global context stroke. There are two categories of context operation:

Global context:

It is the operation which is 2.5 times larger in length than the prediction and whose aggregate local sample ID difference is more than two figure 13. Secondly it has the aggregate samples local ids difference more than 2.0.

Local context:

It is the operation which is 1.5 – 2.0 times larger in length than the prediction and whose aggregate local sample ID difference is less than two. While mapping the prediction to local context where current local context is larger than the previous local context the offsets are applied to the context sample with least difference between the sample local ids that is those samples whose relative position within the operation is similar to the current predicted operation. For instance, in figure 14 first blue sample and second green sample from the bottom have close local sample IDs.

Drawing stage:

Draw the final prediction on the canvas figure 8(a), (b) and (c).

It can be seen that second and third step in our algorithm are unique and contributes to the non-homogeneous constraints mapping of the predictions.

D. Implementation

Neighborhood:

We have set the spatial window figure 2 (f) to 0.05 for fast results. Moreover, we have employed a mechanism of soft constraints analysis which makes predictions more robust based on partial spatial proximity between the samples in the operations as can be seen in 8(c).

Search window:

Spatial window is kept at 0.05 and temporal window 25 operations to accommodate enough operations for obtaining good predictions.

Matching:

While searching for the operation with least similarity we first search the operations in history as defined in temporal window and then we check the spatial matching of the subsequent operations. We set the value of the soft constraints as 0.55.

Spatial & temporal thresholds:

We set the spatial and appearance thresholds to 1.0 and 0.25 figure 2 (k), (l) for better operation search results.

Optimization:

In this work we have skipped the repetitive predictions for current sample at a time to make the process faster and optimal implementation wise.

Sample parameters:

In equation 9 ∂ has 0.1 value and ß has 1.0 value which exhibits more temporally coherent decision which favors good results.

Quality control:

We have noticed that repeating 9, 8 and 10 steps in the above algorithm yields good result but slows the algorithms real-time performance that is why we only stick to 3 iterations.

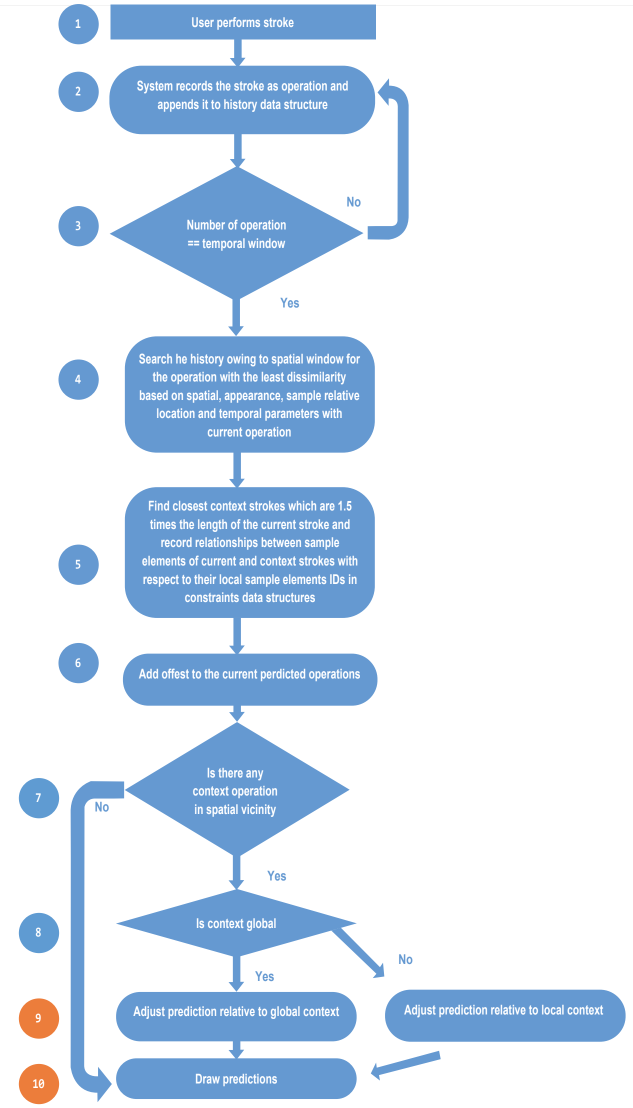

1
User performs stroke
2
System records the stroke as operation and appends it to history data structure
3
Number of operation == temporal window
No
Yes
4
Search he history owing to spatial window for the operation with the least dissimilarity based on spatial, appearance, sample relative location and temporal parameters with current operation
5
Find closest context strokes which are 1.5 times the length of the current stroke and record relationships between sample elements of current and context strokes with respect to their local sample elements IDs in constraints data structures
6
Add offest to the current perdicted operations
7
Is there any context operation in spatial vicinity
No
Yes
8
Is context global
Yes
No
9
Adjust prediction relative to global context
Adjust prediction relative to local context
10
Draw predictions

Fig. (15). Flowchart for our system.

VI. RESULTS

Results for apple shading example figure 3-7 are as follows:

Table 1. Number of strokes comparison for drawings

Figure	Total number of strokes drawn by artist	
	Manual Strokes	Auto completion
3	100	34
4	76	37
5	64	29
6	90	35
7	12	3

[a.] (These results are based on 9 people feedback among which 2 were professional artists)

Table 2. Time comparison for strokes drawn by the artist

Figure	**Time taken for drawing the figure**	
	Time for manual strokes in minutes	**Auto completion**
3	6.4	4
4	4	2.5
5	7.3	3.75
6	8	4
7	4.55	1.25

These results have been compiled from the feedback from 9 users who used our algorithm among which 2 were professional artists and the respective taken by the artists are averaged as can be seen in the table above it can be seen that auto completion provides moderately good results for similar visual quality of the drawing.

VII. DISCUSSION

It can be implied from the results that this auto completion system contributes to saving time and effort on artists.

VIII. CONCLUSION

It can be concluded that there was a moderate improvement with regard to the non-homogeneous context mapping and time taken for predictions related to auto completion while maintaining the same aesthetic quality of the drawing. This work can be improved further by adhering to much larger dataset with advance machine learning techniques like convolutional neural networks.

IX. FUTURE WORK

There is much more vacuum in the field of autocomplete drawing repetition as can be seen from this work the quality of prediction can be improved with more sophisticated techniques like machine learning like [10] used in hand writing recognition recently and [11].

X. ACKNOWLEDGEMENTS

We would like to pay our gratitude to the users and artists who participated in compiling the results and to all the anonymous reviewers.

REFERENCES

[1] R. H. Kazi, F. Chevalier, T. Grossman, S. Zhao and G. Fitzmaurice, “Draco: Bringing life to illustrations with kinetic textures,” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI’14)*. 2014, pp: 351–360.
Doi: 10.1145/2556288.2556987

[2] J. Xing, H-T. Chen and L-Y. Wei, “Autocomplete painting repetitions,” *ACM Transactions on Graphics,* vol. 33, no. 6 (Nov.), pp: 172:1– 172:11, 2014
Doi: 10.1145/2661229.2661247

[3] C. L Zitnick, “Handwriting beautification using token means”. *ACM Transactions on Graphics,* vol. 32, no. 4, pp: 53:1–53:8, 2013.
Doi: 10.1145/2461912.2461985

[4] M. Luka cˇ, J. Fisˇer, J-C. Bazin, O. Jamrisˇka, A. Sorkine- Hornung, and D. Sy kora, “Painting by feature: Texture boundaries for example-based image creation,” *ACM Transactions on Graphics.* vol. 32, no. 4 (July), pp: 116:1–116:8, 2013.
Doi: 10.1145/2461912.2461956

[5] B. Julesz, “Visual Pattern Discrimination,” *IRE Transactions on Information Theory,* vol. 8, no. 2, pp: 84-92, 1962. Doi: 10.1109/TIT.1962.1057698

[6] C. Ma, L-Y. Wei, S. Lefebvre and X. Tong, “Dynamic element textures,” *ACM Transactions on Graphics.* vol. 32, no. 4, 2013. Doi: 10.1145/2461912.2461921

[7] Y. J. Lee, C. L. Zitnick and M. F. Cohen, “ShadowDraw: Real-time user guidance for freehand drawing,” *ACM Transactions on Graphics.* vol. 30, no. 4 (July), pp: 27:1–27:10, 2011. Doi: 10.1145/2010324.1964922

[8] M. Nancel, and A. Cockburn, “Causality: A conceptual model of interaction history,” In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI ’14),* 2014, pp: 1777–1786.
Doi: 10.1145/2556288.2556990

[9] J. Portilla and E. P. Simoncelli, “A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients,” *International Journal of Computer Vision,* vol. 40, no. 1, pp: 49–70, 2000.

[10] K. Gregor, I. Danihelka, A. Graves, D. Rezende and D. Wierstra, “DRAW: A Recurrent Neural Network for Image Generation,” In *Proceedings of the 32nd International Conference on Machine Learning (PMLR)*, 2015, vol. 37, pp: 1462-1471.

[11] L. A. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks,” eprint -arXiv:1505.07376, 2015.