



## Содержание

Вступление . . . . .	4
1 Аналитический раздел . . . . .	5
1.1 Предыдущие работы . . . . .	5
1.1.1 Suggestive Sketching . . . . .	5
1.1.2 Data-driven Sketching . . . . .	5
1.1.3 Using Workflows . . . . .	5
1.2 User Interface . . . . .	6
1.2.1 Future Prediction . . . . .	6
1.2.2 Edit Propagation . . . . .	6
1.2.3 Workflow Clone . . . . .	6
1.3 Method . . . . .	6
1.3.1 Method Overview . . . . .	6
1.3.2 Workflow Texture . . . . .	6
1.3.2.1 Sample . . . . .	6
1.3.2.2 Operation . . . . .	6
1.3.2.3 Neighborhood . . . . .	6
1.3.3 Workflow Synthesis . . . . .	6
1.3.3.1 Initialization . . . . .	6
1.3.3.2 SearchStep . . . . .	6
1.3.3.3 Assignment Step . . . . .	6
1.3.3.4 Weighting . . . . .	6
1.3.4 Workflow Analysis . . . . .	6
1.3.4.1 Local Analysis . . . . .	6
1.3.4.2 Global Analysis . . . . .	6
1.3.4.3 Prediction Quality . . . . .	6
1.3.4.4 Prediction . . . . .	6
1.3.5 Implementation . . . . .	6
1.4 User Evaluation . . . . .	6
1.4.1 Study Protocol . . . . .	6
1.4.2 Target Sketch Performance . . . . .	6
1.4.3 Open Sketch Results . . . . .	6
1.5 Limitations and Future Works . . . . .	6
2 Конструкторский раздел . . . . .	7
2.1 Общая архитектура приложения . . . . .	7
3 Технологический раздел . . . . .	8
3.1 Выбор языка программирования . . . . .	8
3.2 Выбор вспомогательных библиотек . . . . .	8

Заключение . . . . .	9
Список использованных источников . . . . .	10

## Вступление

Повторение является неотъемлемой частью природы, проявляющейся в общих явлениях, таких как поверхностные узоры (например, стены, ткани, полы), геометрические структуры (например, галька, ветви), динамические движения (например, турбулентность жидкости, ходовые циклы, движение толпы) и деятельность человека (Например, рисование, жестикуляция, моделирование). Повторение было важным предметом изучения инженерных и научных дисциплин из-за повсеместности. Главная задача заключается в разработке общих и эффективных методов и простых в использовании интерфейсах для различных явлений и областей приложений. Этот алгоритм представляет собой интерактивную системы для анализа и синтеза повторений эскизов художника. Будет создана интерактивная система цифровой живописи для автодополнения утомительных повторений, таких как штриховки и пунктиры, сохраняющая при этом новаторские вариации и сохраняющая естественные потоки. В отличие от предыдущих работ, посвященных статическим и конечным штрихам, система анализирует рабочий процесс динамического и промежуточного рисования, который позволяет системе понять, как штрихи рисуются в прошлом, чтобы обеспечить высококачественные подсказки, ориентированные на контекст. Пользователи могут рисовать с системой как обычно, в то время как система автоматически предоставляет и обновляет предложения интерактивно без каких-либо дополнительных действий. Пользователи могут игнорировать или принимать эти предложения, аналогичные функциям автозаполнения в интегрированных средах разработки программирования, тем самым сохраняя полный контроль над процессом рисования.

# **1 Аналитический раздел**

## **1.1 Предыдущие работы**

### **1.1.1 Suggestive Sketching**

Drawing is a common and yet challenging human activity. Significant research efforts have been devoted to design suggestive or guided drawing systems that use various forms of data to help users. For example, portrait sketching can be assisted by crowd-sourced sketches [1] or analyzed face data [1]. To help users draw a larger collection of objects, Lee et al. [1] interactively guides users' progress by displaying shadows extracted from web images, while Iarussi et al. [1] provide structural guidances based on artistic and geometric principles. Besides static images or drawings, it is also possible to guide novices through recorded workflows of experienced painters [1]. Our method follows this line of work, but uses past drawing workflows from the same user to guide future drawings.

### **1.1.2 Data-driven Sketching**

Although the guided systems can help users draw easier, they still have to deal with manual repetitions. To help ameliorate such tedious process, many systems have been designed to automate the authoring of repetitive drawings via data driven computation. One prime example is the detailed textures or patterns [1] [1] [1] [1] whose creation can fit well with the traditional copy-and-paste interaction styles. Kazi et al. [1] [1] provide friendly gesture controls to help create both static and dynamic elements. Our method also follows a data-driven approach to help creating repetitions, but uses dynamic workflows rather than static patterns or animated sprites.

### **1.1.3 Using Workflows**

Chimera [1] is an old system for recording and editing graphical histories, specifically the repetitive operations. Recent years have seen the rise of methods that utilize workflows in various forms, such as exploration [1], visualization [1] [1] [1] [1], stylization [1], beautification [1], tutorial [1] [1], and revision [1]. These methods often rely on pre-recorded workflows, which, when not available, may also be recovered through a certain extent of analysis [1] [1] [1] [1]. Recently, Nancel et al. [1] provides a comprehensive survey of different conceptual models for workflow analysis. We follow this line of work, but focus on the analysis and synthesis of dynamic workflows for suggestive painting repetitions.

- 1.2 User Interface
  - 1.2.1 Future Prediction
  - 1.2.2 Edit Propagation
  - 1.2.3 Workflow Clone
- 1.3 Method
  - 1.3.1 Method Overview
  - 1.3.2 Workflow Texture
    - 1.3.2.1 Sample
    - 1.3.2.2 Operation
    - 1.3.2.3 Neighborhood
  - 1.3.3 Workflow Synthesis
    - 1.3.3.1 Initialization
    - 1.3.3.2 SearchStep
    - 1.3.3.3 Assignment Step
    - 1.3.3.4 Weighting
  - 1.3.4 Workflow Analysis
    - 1.3.4.1 Local Analysis
    - 1.3.4.2 Global Analysis
    - 1.3.4.3 Prediction Quality
    - 1.3.4.4 Prediction
  - 1.3.5 Implementation
- 1.4 User Evaluation
  - 1.4.1 Study Protocol
  - 1.4.2 Target Sketch Performance
  - 1.4.3 Open Sketch Results
- 1.5 Limitations and Future Works

## 2 Конструкторский раздел

### 2.1 Общая архитектура приложения



Рисунок 2.1 — Общая архитектура приложения

## **3 Технологический раздел**

### **3.1 Выбор языка программирования**

Для реализации был выбран язык C++ по следующим причинам:

- а) The quick brown fox jumps over the lazy dog.
- б) Компилируемый язык со статической типизацией.
- в) Сочетание высокоуровневых и низкоуровневых средств.
- г) Реализация ООП.
- д) Наличие удобной стандартной библиотеки шаблонов

### **3.2 Выбор вспомогательных библиотек**

Для реализации программы была выбрана библиотека Qt.

- а) The quick brown fox jumps over the lazy dog.
- б) The quick brown fox jumps over the lazy dog.



## Заключение

## **Список использованных источников**

1. *Author.* Title / Author. — Publisher, 2007.