CS201: Data Structures II Project Proposal

Project Title: Versionary: An Image-Based Version Control System
Team Members:
1. Azfar Ali - aa08861
2. Minhaj ul Hassan - mu08984
3. Muhammad Qasim Khan - mk05539
Instructor: Dr. Usman Arif

March 17, 2025

1 Abstract

Versionary is an image-based version control system designed to efficiently manage and track changes between image versions. The core of the system relies on Merkle Trees to ensure secure and efficient version tracking by storing image hashes. By converting images to grayscale, Versionary reduces the amount of data to process, making the system faster and more efficient. Additionally, Quadtrees are used to break down images into smaller chunks to optimize the comparison process. This combination allows Versionary to detect differences between versions, even in large images, making it an effective tool for image version control and visual quality assurance.

2 Introduction

Managing image versions is crucial in fields like graphic design, geospatial analysis, and visual monitoring. Traditional version control systems, such as Git, are primarily designed for text-based changes and are not well-suited for handling images. Versionary addresses this gap by providing a dedicated system for image versioning and change tracking. At its core, Versionary leverages Merkle Trees to maintain secure and efficient version tracking, ensuring data integrity through cryptographic hashing. To further enhance efficiency, the system employs grayscale conversion to simplify processing and uses Quadtrees to segment images for localized comparisons. This combination makes Versionary a practical solution for real-world image version management.

3 Objectives

- Develop a system to efficiently track changes between image versions.
- Leverage Merkle Trees for secure and efficient version tracking.
- Use grayscale images to reduce processing complexity and improve speed.
- Use Quadtrees as a supportive structure to manage image differences effectively.
- Implement features like version rollback and difference detection.
- Create a simple and user-friendly command-line interface (CLI) for version management.

4 Methodology

- **Merkle Tree Construction:** Store image hashes in a Merkle Tree to maintain integrity and support efficient version comparison.
- **Grayscale Conversion:** Convert images to grayscale to simplify processing.
- **Data Structures:** Use Merkle Trees as the primary structure and Quadtrees as supportive structures to segment images.

- **Hashing Algorithm:** Use SHA-256 for secure and efficient hash computation.
- **Image Comparison:** Compare grayscale hashes to find differences between versions.
- **Implementation:** Use C++ for core functionality and OpenCV for image processing.

5 Expected Outcomes

Versionary will streamline image versioning and change tracking, offering features like version rollback, rapid difference detection, and optimized storage management. By utilizing Merkle Trees as the backbone for secure version management and incorporating Quadtrees for efficient image segmentation, Versionary will serve as a powerful tool for applications such as graphic design, quality monitoring, and geospatial data analysis.

6 References

- Merkle Trees and Cryptographic Hash Functions - `https://en.wikipedia.org/wiki/Merkle_tree`
- Quadtrees: Efficient Spatial Indexing - `https://en.wikipedia.org/wiki/Quadtree`
- SHA-256 Hashing Algorithm - `https://en.wikipedia.org/wiki/SHA-2`
- OpenCV: Computer Vision Library - `https://opencv.org/`