

Abstract

BACKGROUND: Choosing the right data structure for representing two-dimensional data, especially when we do not know the computer memory model, is an important aspect. There are also a number of problems where we need to roll back to previous versions of the data structure. It is not a big problem to find an implementation of such data structure in an imperative programming language, but it becomes difficult if we want to solve such a problem in a functional language.

OBJECTIVE: This study investigates purely functional Quadtree that can be used in both persistent and ephemeral contexts. Also, such data structure does not rely on the characteristics and memory size of the computer, which makes it cache-oblivious.

METHODS: The data structure was implemented in the Haskell programming language. A hybrid data structure that uses a packed memory array and Edward Kmett's map was used as a fundament of the Quadtree. Additionally, we developed a visual interface for visualising the work of data structure.

RESULTS: The solution shows speed and memory gains compared to existing data structures. It makes it possible to effectively solve problems in functional programming languages.

CONTRIBUTION AND APPLICABILITY: This data structure is particularly useful when dealing with two-dimensional data and unknown computer memory parameters. It offers the advantage of being able to roll back to previous versions of the data without additional memory costs.