

## Mapeamento de Informações Geomorfológicas para Ambientes Computacionais

Guilherme Henrique S. Arruda<sup>1</sup>, Sandro Sawicki  
UNIJUI, Departamento de Ciências Exatas e Engenharias  
Rua do Comércio, 3000. 98700-000, Ijuí – RS – Brasil  
guilherme\_arruda@live.com, sawicki@unijui.edu.br

*Palavras-chave:* Mapeamento Geomorfológico; Fast Light Tool Kit.

Nesta nova etapa do projeto, foi dado início aos passos que foram destacados na última apresentação. As funções desenvolvidas foram fundamentais para a utilização de outros métodos de detecção, obtendo objetos, áreas e formas diferentes dos que estavam sendo analisados no princípio da aplicação. A razão da busca por novos objetos surgiu devido às áreas utilizadas serem muito amplas, o que facilitava a detecção de contornos, mas dificultava a adição do obstáculo na aplicação de cobertura de sinais devido ao tamanho da imagem resultante e da quantidade de coordenadas obtidas. Isso apresentou resultados pouco satisfatórios, pois, com áreas mais distantes e objetos menores, os contornos ficavam muito próximos uns dos outros, tornando a saída irregular.

A partir disso, surgiu a necessidade de reorganizar a aplicação no que diz respeito à sua estrutura. As funções e variáveis desta foram desenvolvidos estruturalmente, tanto por uma questão de testes, pois se tratava de uma biblioteca nova, quanto pelo motivo de que a organização não era prioridade no início do projeto. Deste modo, o poder do C++ não estava sendo aproveitado. Toda a parte estrutural foi então dividida em classes, facilitando a identificação dos processos e separando algumas partes do projeto, objetivando o encapsulamento. A orientação a objetos também foi adicionada, que é outro recurso poderoso do C++.

Outra mudança positiva no projeto foi a adição de interface gráfica para facilitar a execução da aplicação. Primeiramente, foi feito um estudo em relação à parte gráfica na linguagem C++. Optou-se por utilizar FLTK (Fast Light Tool Kit) pelo fato de ser simples e fácil de utilizar, compatível com OpenGL e possuir funções que se adaptavam ao nosso projeto. Assim, a parte gráfica foi pré-modelada e então desenvolvida com os recursos disponíveis. A interface possui um explorador de arquivos, onde é possível escolher graficamente uma imagem a ser manipulada.

Ao fazer isso, uma segunda janela é aberta, e a imagem é carregada em um espaço próprio, sendo este integrado com ferramentas para manipulá-la. As funções, que antes precisavam ser escolhidas via comando através do terminal, agora possuem seu próprio botão. Quando um destes é pressionado, os efeitos na imagem são executados em tempo real, o que facilita muito a identificação das mudanças ocorridas. A adição da parte gráfica

---

<sup>1</sup> Bolsista PIBIC/UNIJUI

possibilitou a integração de funções, o que antes não era possível fazer porque apenas uma função de cada vez poderia ser executada através do terminal. Esta integração tornou o processo de obtenção dos obstáculos mais fácil e preciso. As coordenadas obtidas podem ser dinamicamente salvas ao pressionar seu botão característico, processo que é feito graficamente.

Portanto, é fácil perceber o quanto a aplicação mudou diante do que foi apresentado. São inúmeros os benefícios resultantes, tanto para o tempo de execução quanto para o resultado final. Os próximos passos serão destinados à adição de mais funções de manipulação de imagem na interface e a integração destas, possibilitando trabalhar melhor com áreas mais afastadas.