

Heurísticas y Método de Punto Proximal para la Reconstrucción Automática de Geometría en Lechos Acuáticos

C. García Bauza^{a,b}, V. Cifuentes^{a,b}, L. Parente^c, P. Lotito^{a,c}, M. Vénere^{a,d}

^aPLADEMA, Universidad Nacional del Centro, 7000 Tandil, Argentina,
([crgarcia](mailto:crgarcia@exa.unicen.edu.ar), [cifunte](mailto:cifunte@exa.unicen.edu.ar), [venerem](mailto:venerem@exa.unicen.edu.ar))@exa.unicen.edu.ar)

^b Comisión de Investigaciones Científicas de la Provincia de Buenos Aires

^c Consejo Nacional de Investigaciones Científicas y Técnicas

^d CNEA

Abstract

This paper presents several algorithms for the automatic river-bed reconstruction. Navigation channel is defined by a scarce set of bathymetry points which are obtained from several cross-sections of the river-bed. At the rendering stage, the scarce depth values and their spatial arrangement generates low-quality polygonal approximations of the real river. In order to preserve realism in the visual representation, a heuristic based on Delaunay and a Proximal Point Method are proposed. Some results obtained for a 400 km cross-section of the Paraná River are included.

Keywords: Triangulation, Topographic Models, Mesh Reconstruction, GIS

Resumen

Se presentan algoritmos para la generación de mallas poligonales a partir de conjuntos de puntos batimétricos correspondientes a canales de navegación, cuya geometría se corresponde con distintos cortes transversales del río. A la hora del renderizado, la escasa información junto a la disposición espacial de las profundidades relevadas originan aproximaciones poligonales con calidad poco aceptable empobreciendo la apariencia visual del lecho real. Heurísticas basadas en la triangulación de Delaunay y la aplicación de métodos de Punto Proximal permitieron otorgar mayor realismo a la representación visual. Se presentan los resultados obtenidos para una sección de 400 Km. del río Paraná.

Palabras Clave: Triangulación, Modelos Topográficos, Reconstrucción de Mallas, GIS

1 INTRODUCCIÓN

En ríos navegables es de vital importancia el disponer de una descripción precisa y actualizada de la profundidad en cada sector del mismo. En general esta información no es estática, ya que permanentemente se realizan dragados, aparecen bancos de arena u ocurren desmoronamientos de costa, lo cual puede cambiar sustancialmente las condiciones para la navegación. Por este motivo en cauces importantes, como es el caso del Paraná o parte del Uruguay, se realizan mediciones continuas, las cuales son luego suministradas a las embarcaciones en forma de una nube de puntos en el espacio [1,2].

Esta información en crudo resulta de poca utilidad si no se la procesa y visualiza adecuadamente. En general se trata de una nube de puntos de distribución no uniforme, que alterna sectores muy densificados junto a otros muy dispersos. Es corriente por ejemplo que la distribución sea un conjunto de cortes transversales del río que agrega la batimetría para la línea media. En particular, el escenario estudiado corresponde aproximadamente a 400 kilómetros del recorrido del Paraná, incluyendo no solo el lecho fluvial, sino también las líneas de costa (Figura 1).

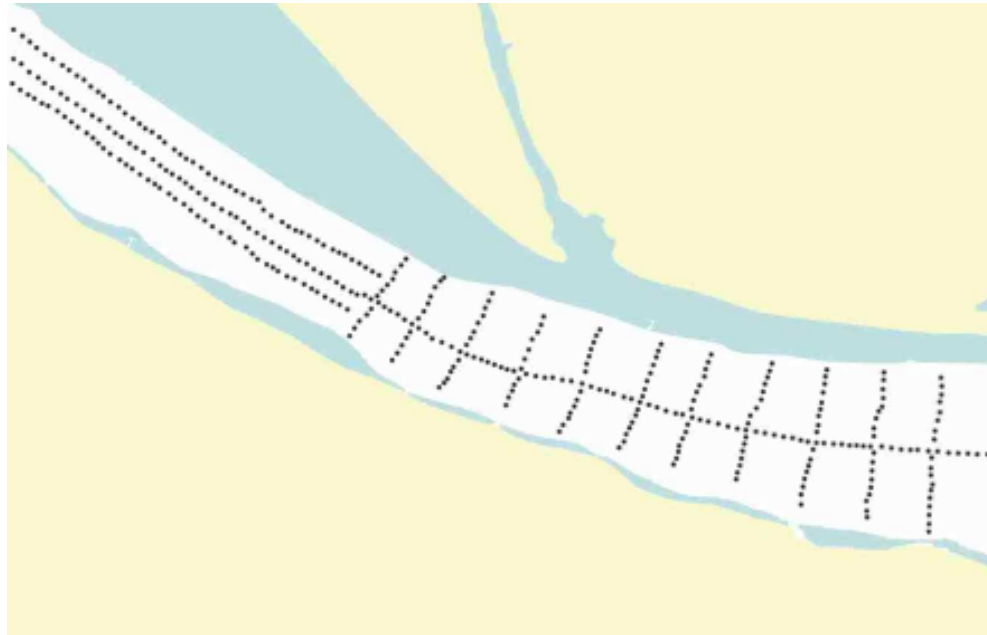


Figura 1. Nube de puntos correspondiente a un sector del relevamiento batimétrico.

Las aplicaciones de software comercializadas en la actualidad completan y/o corrigen la información faltante de los modelos digitales de elevación con estrategias escasamente documentadas y funcionamiento desconocido [3,4]. En este trabajo se presentan varios algoritmos que permiten construir en forma semi-automática triangulaciones suaves del dominio de interés.

En primer lugar se generó la superficie basados en la triangulación de Delaunay del conjunto de puntos. En general esto genera triangulaciones de muy mala calidad que no pueden utilizarse en forma directa para una visualización [5]. Posteriormente se utilizaron varias heurísticas para incorporar puntos adicionales y lograr así una superficie más suave y triángulos mas proporcionados. Tales modificaciones involucran principalmente el agregado de nuevos puntos, dividiendo aquellos triángulos cuya magnitud de área supera una dada tolerancia.

Si bien las heurísticas propuestas construyeron triangulaciones plausibles del lecho acuático en estudio, igualmente se implementó la reconstrucción a partir de un problema de extensión Lipschitziana optimal. De modo que, el problema original se discretiza para transformarlo en un problema de optimización no lineal y no diferenciable, en dimensión finita, que es resuelto numéricamente utilizando métodos de punto proximal. En particular, la información tratada representa a la profundidad en puntos fijos de una grilla, y se deben elegir los valores de los puntos restantes minimizando la norma L-infinito del gradiente calculado mediante la aproximación en diferencias finitas. Los resultados obtenidos con esta técnica son más que interesantes, ya que genera superficies suaves, sin grandes polígonos que distorsionan la vista.

2 REDES DE TRIÁNGULOS IRREGULARES A PARTIR DE CONTORNOS

En un principio se aplicó uno de los algoritmos que permiten obtener la triangulación de Delaunay en 2D de un conjunto arbitrario de puntos. El mismo se basa en el agregado secuencial de nuevos puntos a una triangulación que ya es Delaunay: Se comienza con dos triángulos que cubren a la totalidad de los puntos y a partir de aquí, en cada paso se incorpora un punto nuevo removiendo aquellos triángulos que fallan el test Delaunay y generando nuevos uniendo los lados de la cavidad generada con el nuevo punto en cuestión. En la figura 2 se ejemplifica este proceso y en [6] se puede encontrar una descripción detallada del algoritmo.

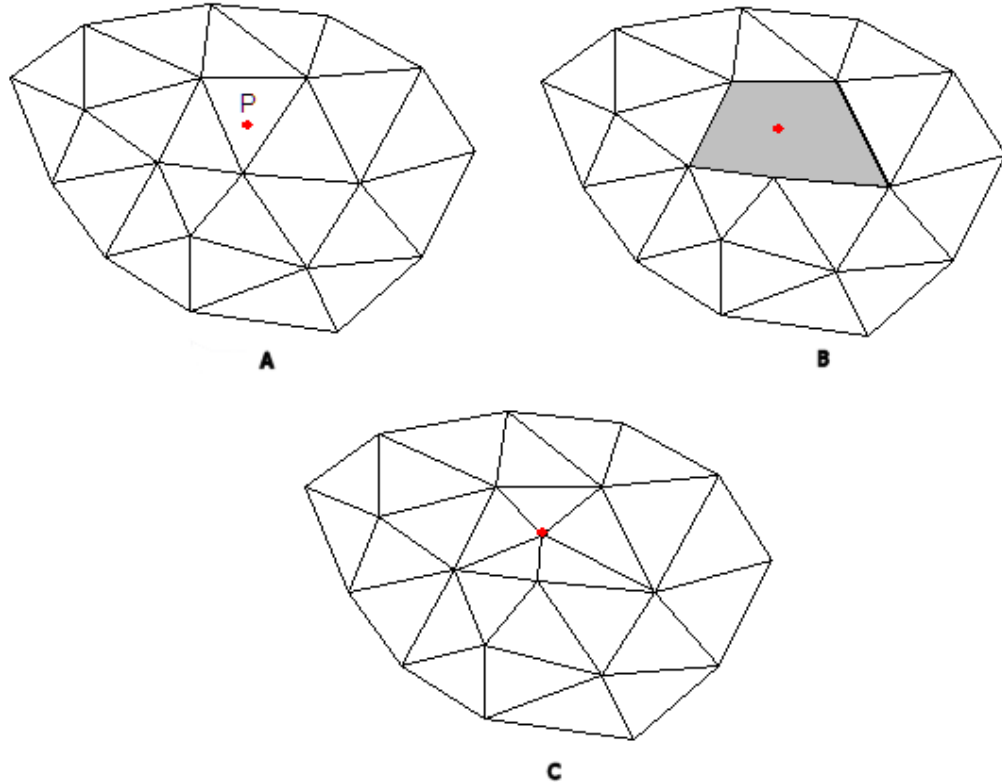


Figura 2. Triangulación de Delaunay: agregado de un punto (A), eliminación de polígonos que no cumplen la condición (B) y retriangulación (C).

De esta forma a partir de la nube batimétrica se genera un modelo poligonal del cauce del río. Como ya se mencionó, tal aproximación resultó poco satisfactoria, obteniéndose mallas como la de la figura 3 muy densificadas en determinadas zonas y con grandes regiones planas entre ellas que resultan poco convincentes con respecto a la superficie que se quiere reconstruir.

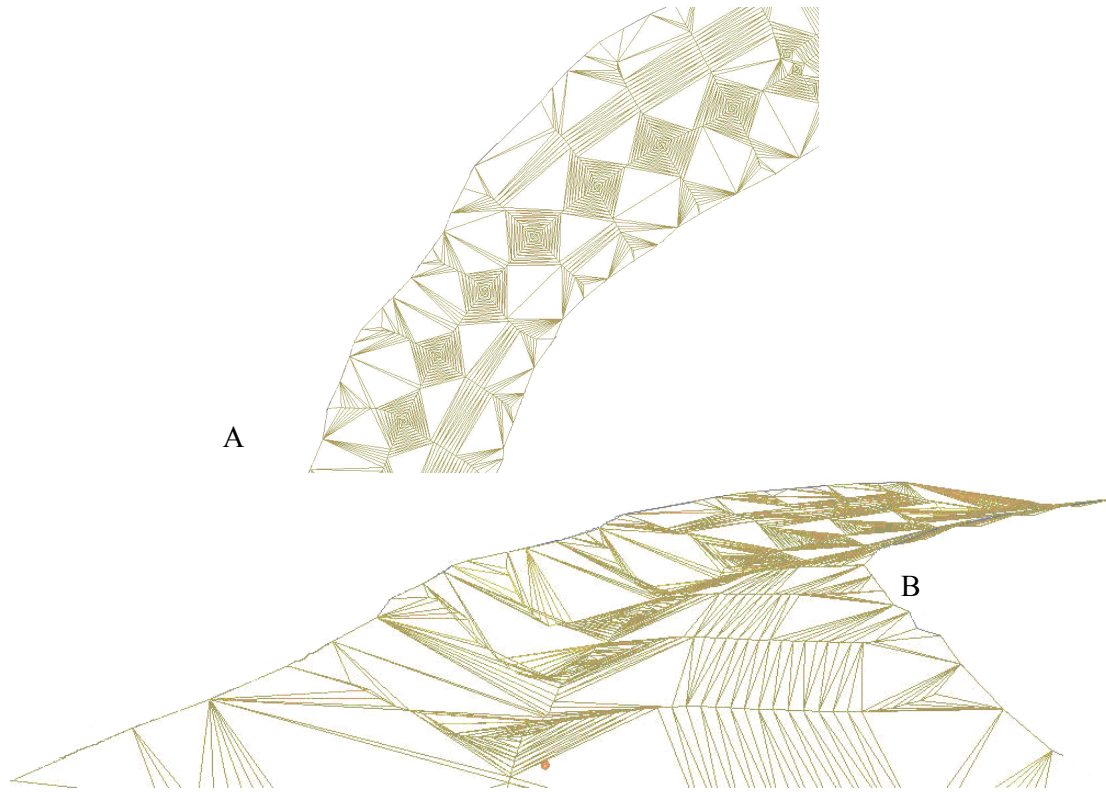


Figura 3 – Vista en planta (A) y en perspectiva (B) de la malla generada con Delaunay modificado.

Uno de los problemas que más molesta, es que dependiendo de la distribución de los puntos se pueden generar algunos triángulos que vinculan en forma directa un punto en la costa con otro en el centro del cauce, generando especies de “diques” y pozos que obviamente no están presentes en la realidad.

3 HEURÍSTICA DE RECONSTRUCCIÓN CON AGREGADO SELECTIVO DE PUNTOS

Se intentó entonces agregar nuevos puntos a la triangulación de forma de minimizar estos problemas. Una de las estrategias fué el agregado de puntos dentro de aquellos triángulos cuya magnitud de área supera una dada tolerancia. En este caso, se agrega selectivamente el centroide del triángulo repitiendo el proceso hasta que todas las áreas de los triángulos incorporados no superen el tolerado formando a su vez nuevos triángulos que cumplan con los criterios mencionados [7].

La utilización de ésta técnica híbrida trajo una solución razonable al problema. Como se puede ver en la figura 4, el agregado selectivo de puntos produce una superficie de apariencia mucho más suave, desapareciendo los saltos producidos por ausencia de información. Un valor de área máxima de 500 m² se usó para obtener un buen equilibrio entre calidad y tiempo de procesamiento.

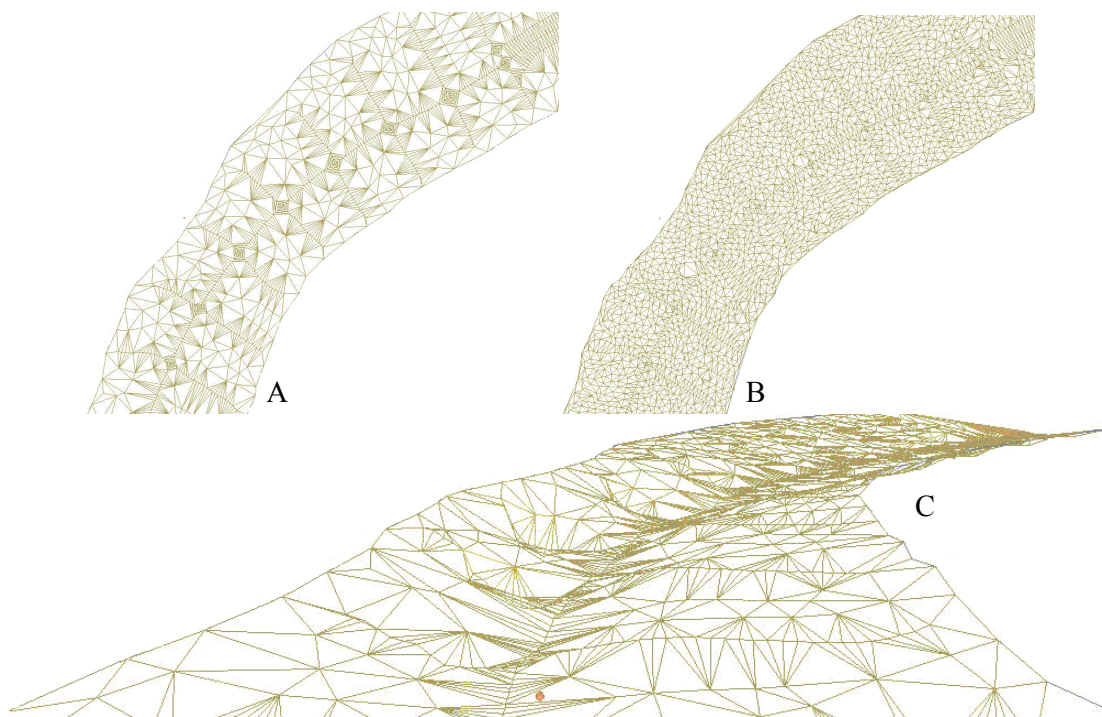


Figura 4 – Triangulación con agregado selectivo de puntos con área de triángulo máxima de:1000 m2 (A) y de 300 m2 (B y C).

4 MÉTODO DEL PUNTO PROXIMAL

Una alternativa diferente para obtener este tipo de superficies es plantearlo como un problema de optimización con restricciones. Por ejemplo se puede buscar la superficie que pasa por los puntos dato con la mínima curvatura, o con menor gradiente. En este trabajo incluimos la experiencia que realizamos basados en un problema de extensión Lipschitziana optimal. Esto es, dado un conjunto de celdas con valores fijos (obtenidos a partir de los datos medidos en nuestro problema) se busca una extensión a un dominio más amplio, que minimice la constante de Lipschitz.

Una función f definida en un subconjunto de R^n , se dice que es **lipschitziana** en x si las variaciones locales de la función están acotadas por una constante multiplicada por la variación de la variable, más precisamente, si existe una constante K tal que para todo valor de y se tiene que

$$|f(x) - f(y)| \leq K|x - y|.$$

La menor cota K que verifique la desigualdad anterior se denomina **constante de Lipschitz**. En el caso de funciones derivables, la constante de Lipschitz es una cota de la norma gradiente, y buscar la función de menor cota de Lipschitz equivale a buscar una función de variación mínima en el sentido del gradiente.

Este problema es la versión discreta del problema general de obtener la función lipschitziana de mínima constante de Lipschitz entre todas las funciones que con valores fijos en un subconjunto de su dominio. Dicho de otra manera, si, por ejemplo, se busca una función definida en $[0, 2]$ y se conocen los valores que toma en 0, 1, y 2, entonces la función buscada es una función cuya constante de Lipschitz no supera la que ponen los valores prefijados, en este caso resulta seccionalmente lineal, ver figura 5. Para funciones definidas en la recta es casi trivial, ya que basta definir la función como lineal (afin) en cada subintervalo. Para el caso de funciones definidas en el

plano, el problema es más interesante ya que por cuestiones topológicas no se puede definir un orden en el dominio de modo de construir la extensión como en el caso anterior (unidimensional).

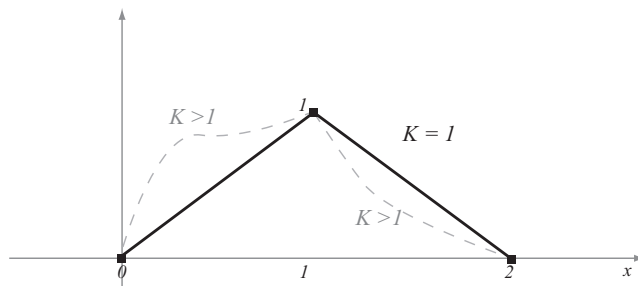


Figura 5 – Ejemplo de extensión lipschitziana unidimensional.

A su vez el problema de extensión Lipschitziana minimal es una caso particular del problema de minimizar un funcional de la forma¹

$$J(u) = \sup_{x \in \Omega} f(x, u, D(u)),$$

definido para u en W , conjunto de funciones lipschitzianas sobre Ω conjunto acotado de R^n . En nuestro caso de dimensión 2, u representa la superficie buscada y $f = \|\nabla u\|$. La existencia de soluciones está garantizada con condiciones bastante débiles sobre la función f , como la quasi-convexidad, véase [10]. El interés de encuadrar nuestro problema en este marco radica en poder utilizar los métodos desarrollados en [10] para su resolución.

Para ello, y siguiendo el esquema presentado en [10], utilizamos un procedimiento de discretización basado en diferencias finitas. Sin pérdida de generalidad podemos suponer que $\Omega = (0,1)^2$. Dado $p \in N$, definimos $k = 1/p$, y consideramos la discretización Ω_k de Ω en celdas de tamaño $1/p \times 1/p$. Para definir la discretización W_k de W , consideramos triangulos superiores e inferiores en $\bar{\Omega}_k$, obteniendo $W_k = \{v \in W, \text{afin en todo triángulo}\}$.

Las funciones v son consideradas como vectores de dimensión $(p+1)^2$ donde las componentes $v_{i,j}$ son libres salvo para las coordenadas donde tenemos datos. Para el problema de reconstrucción, a partir de las coordenadas donde se efectuaron mediciones se genera una grilla cuadrada y se identifican las celdas que incluyen mediciones. Cada celda se corresponderá con una componente de v , siendo fijas las componentes asociadas a celdas que incluyan mediciones. En el caso de que haya más de una medición en una celda se asignará el promedio de los valores medidos en la celda correspondiente. Como $W_k \subset W$, tenemos que $\inf_{u \in W} J(u) \leq \inf_{v \in W_k} J(v)$.

Las derivadas parciales discretizadas se definen en la forma usual y utilizamos combinaciones convexas de estas diferencias parciales para definir el gradiente de v en puntos vecinos al borde. Una vez que las variables están discretizadas, definimos $F^{i,j}$, la discretización de la función $f(x, v, Dv)$ en los nodos $x_{i,j}$, como

$$F^{i,j}(v) = f(x_{i,j}, v_{i,j}, (\nabla v)_{i,j}), i, j = 0, \dots, p.$$

¹ La definición correcta es $J(u) = \text{ess sup}_{x \in \Omega} f(x, u, D(u))$, para $u \in W_g^{1,\infty} = \{v \in W^{1,\infty} : v = g, \text{sobre } \partial\Omega\}$, ver [10].

Además, definimos el funcional discreto en $W_k : J_k(v) = \max_{i,j=0,\dots,p} F^{i,j}(v)$. Se puede ver en [8] que la solución del problema discretizado converge a la del problema continuo si se reduce la malla, es decir que

$$\lim_{k \rightarrow \infty} [\inf \{J(v) : v \in W\} - \inf \{J_k(v) : v \in W_k\}] = 0.$$

El problema a ser resuelto numéricamente resulta

$$\min_{v \in R^p} \max_{i,j=0,\dots,p} F^{i,j}(v).$$

Notemos que $J_k(v) = \max_{i,j=0,\dots,p} F^{i,j}(v)$ es un funcional no diferenciable convexo para funciones

convexas $F^{i,j}$, por lo tanto admite subdiferencial y el problema es equivalente a hallar v tal que $0 \in \partial J_k(v)$, donde ∂J es el subdiferencial de J , que generaliza la noción de gradiente al caso no diferenciable. Para tratar este problema de optimización convexa no diferenciable se puede considerar algún algoritmo de tipo “haces” como los presentados en [9]. Dentro de dicha clase utilizaremos el Método de *Punto Proximal Inexacto Híbrido con Métrica Variable*, presentado en [11].

Dicho método posee todas las propiedades deseables de convergencia del Método de Punto Proximal introducido por Rockafellar en [12], pero el criterio de aproximación es considerablemente más constructivo y aplicable que las condiciones clásicas de aproximación del tipo sumabilidad de Rockafellar. La implementación de métrica variable es útil para acelerar la convergencia y para evitar problemas numéricos relacionados con cuestiones de escala.

El hecho de sumergir la superficie en una grilla cuadrada hace que se agregue una cantidad importante de variables innecesarias que reducen significativamente la velocidad de convergencia del método numérico. Para poder eliminarlas se necesita un procesamiento que en esta primera etapa no se realizó. Es por ello que en la figura 6, sólo presentamos la reconstrucción de una sección del río.

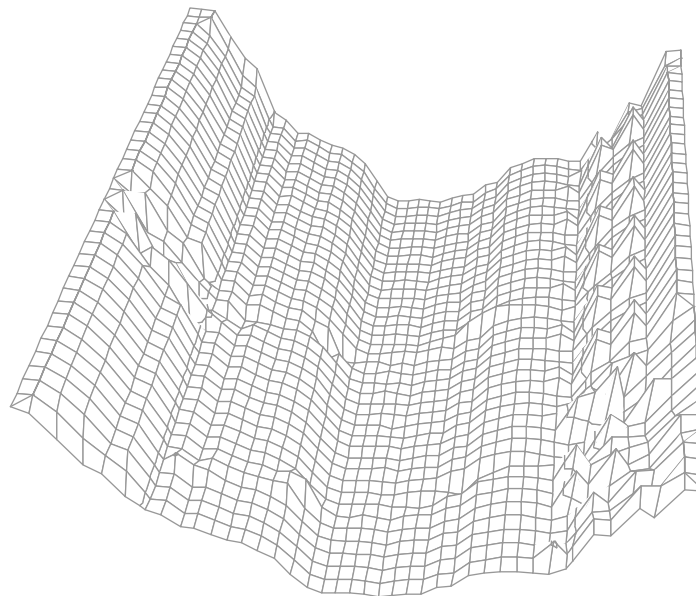


Figura 6 – Reconstrucción del lecho Acuático basado en el método del punto proximal.

Como puede observarse, este método genera superficies mucho más adecuadas y suaves que los algoritmos basados en triangulaciones Delaunay.

5 CONCLUSIÓN Y TRABAJOS FUTUROS

Se implementaron y analizaron una serie de algoritmos para obtener representaciones de superficies cuando solo se dispone de algunos puntos de la misma. Por un lado los algoritmos basados en el método Delaunay resultan sumamente robustos y económicos, pero la calidad de las superficies que se obtienen no son todo lo satisfactorias que se podría desear.

La idea de utilizar métodos de optimización como una alternativa parece sumamente prometedora. Las pruebas realizadas sobre sectores reducidos de los datos disponibles muestran resultados más que alentadores. Resta trabajar en mejorar su costo computacional para poder aplicarlo en forma directa al set completo de datos, para lo cual se puede utilizar un esquema de grilla no regular para su resolución por diferencias finitas y que incluya estrictamente al lecho de interés (en las pruebas realizadas se abarca un sector rectangular).

REFERENCIAS

- [1] Prefectura Naval Argentina. Descarga y consulta de datos en <http://www.prefecturanaval.gov.ar/institucional/castellano/index.htm>
- [2] Asociación Civil de Prácticos del Río Paraná. <http://pilots-rioparana.com/>
- [3] 3DEM Software. Sitio web: <http://www.visualizationsoftware.com/3dem>.
- [4] SRTMFILL. Sitio web: <http://www.3dnature.com/srtmfill.html>.
- [5] Gold C. M y Dakowicz M. Terrain modeling based on contours and slopes. *D. Richardson and P. van Oosterom, editors, Advances in Spatial Data Handling. Proceedings, 10th International Symposium on Spatial Data Handling*. Springer-Verlag Berlin, 2002.
- [6] Venere M. y Dari E. Análisis comparativo de algoritmos para obtener triangulaciones Delaunay. *Mecánica Computacional*, v.10, 491-506, 1990.
- [7] Nigel R Weatherill. Delaunay Triangulation in Computational Fluid Dynamics. *Computers and Mathematics with Applications* 24(5/6):129-150, 1992.
- [8] Aragone L., González R.L.V. y Reyero G. Numerical approximation of optimization problems with L^∞ functionals. *Investigación Operacional*, Cuba, Nro. 1-04, v.25, pp. 14-24. ISSN 0257-4306, 2004.
- [9] Bonnans J. F., Gilbert J. C., Lemarchal C., Sagastizbal C. A., *Numerical Optimization*. Springer, 2003.
- [10] Parente L. A., Aragone L. S., Lotito P. A. y Reyero G. F. Numerical solution of a variational problem with L^∞ functional. *Proceedings in Applied Mathematics and Mechanics*, v.7, Issue 1, pp. 1060401-1060402, ISSN: 1617-7061, 2007.

- [11] Parente L. A., Lotito P. A. y Solodov M. V. A Class of Inexact Variable Metric Proximal Point Algorithms. *SIOPT, SIAM Journal on Optimization*, v.19, pp. 240-260, ISSN: 1052-6234, 2008.
- [12] Rockafellar R. T. Monotone Operators and the Proximal Point Algorithm. *SIAM Journal of Control and Optimization*, v.14, pp. 877-898, 1976.