

Planteamiento y avances del
PROYECTO DE INVESTIGACIÓN

Determining the position of a mobile robot by means of the spectral
signature of images.

José Ignacio López Peña

Tutor: *Ángel Fernando Kuri Morales*

Tabla de contenidos

1	RESUMEN	2
2	OBJETIVO	4
3	ANTECEDENTES	5
4	ANÁLISIS DEL ESTADO DEL ARTE	6
5	METAS	7
6	METODOLOGÍA	8
6.1	OPTIMIZACIÓN MULTI-OBJETIVO	8
6.2	MÉTODO DEL CEDAZO (SIEVE METHOD)	9
6.3	OPTIMIZACIÓN MULTI-OBJETIVO BASADO EN LA NORMA L_{∞}	11
6.4	ORB.....	16
6.5	PRE-PROCESAMIENTO DE LOS DATOS	17
6.6	APROXIMACIÓN MULTIVARIADA.....	22
6.6.1	<i>Algoritmo de Levenberg-Marquardt</i>	24
6.6.2	<i>Algoritmo de Powell</i>	24
6.6.3	<i>Algoritmo de Rápido Ascenso</i>	24
6.7	POLINOMIOS GENÉTICOS MULTIVARIADOS.....	24
6.8	REDES DE PERCEPTRONES MULTICAPA (MLPs)	27
6.8.1	<i>Determinación de la arquitectura de las MLPs</i>	29
7	RESULTADOS	30
8	SIGUIENTES PASOS	36
9	CONCLUSIONES Y DISCUSIÓN	37
10	PUBLICACIONES A LA FECHA	37
11	REFERENCIAS	38

1 Resumen

En el presente trabajo se plantea la solución al problema de identificación de la posición de un robot móvil dentro de un ambiente conocido identificando las frecuencias correspondientes al conjunto óptimo de los puntos de las imágenes previamente obtenidas y almacenadas. En vez de analizar características aisladas o predefinidas en dicho conjunto de imágenes, hacemos un análisis holístico de frecuencias. Se discute la metodología que permite resolver cada uno de los problemas que hay que salvar para lograr este objetivo.

Para lograr lo anterior, el primer paso consiste en identificar el subconjunto óptimo de píxeles que permitan al robot determinar su posición. Tácitamente está implícito resolver un problema que involucra dos objetivos que entran en conflicto (lo que constituye un problema de tipo multi-objetivo): a) por un lado requerimos determinar el número mínimo de puntos necesarios que caractericen el ambiente y b) por otro queremos que el error de identificación sea el menor posible. En un trabajo previo (Maestría) [42] resolvimos este problema de forma preliminar utilizando el método del Cedazo sobre un conjunto de imágenes en el espacio de color RGB. El método del Cedazo se basa en un Algoritmo Evolutivo Multi-Objetivo (MOEA). Este arroja como resultado un vector de posibles soluciones, llamado soluciones óptimas de Pareto, de las cuales debemos seleccionar una. Durante el presente proceso de investigación nos dimos cuenta que se puede reemplazar el MOEA utilizado en el método del Cedazo original, por un algoritmo más poderoso basado en la norma L_∞ ¹. Partiendo de la base de que ya existen algoritmos que resuelven problemas multi-objetivo realizamos una comparación experimental. Como método base de nuestra comparación seleccionamos el algoritmo MOSHERPA. Sus autores demostraron que es el mejor para resolver de optimización multi-objetivo [7] resolviendo un conjunto de funciones de referencia especialmente diseñadas para probar este tipo de algoritmos [63]. Nosotros realizamos una prueba análoga utilizando la norma L_∞ sobre el mismo conjunto de funciones. Nuestros resultados muestran que nuestro algoritmo siempre encuentra un punto que pertenece a las soluciones óptimas de Pareto. Esto ocurre en pocas generaciones y con un costo computacional considerablemente menor que el reportado para MOSHERPA (y otros algoritmos MOEA).

Para identificar las imágenes, en [42] se utilizaron diferentes medidas de similitud basadas en la distancia métrica de Minkowsky². Sin embargo es posible utilizar técnicas de aprendizaje supervisado para lograr este objetivo. En este sentido el problema consiste en encontrar un modelo que describa lo que sabemos y que tenga capacidad de generalizar para que también describa lo que no sabemos. Dicho de otra forma, deseamos encontrar el modelo matemático del sistema de la forma $y = f(\mathbf{v})$. El modelo que nosotros que buscamos es una expresión de una variable dependiente y (la imagen a identificar) en función de un conjunto

¹ La norma L_∞ define como $\|\mathbf{x}\|_\infty = \max_i (\|x_i\|)$ [61] donde x_i son las funciones objetivo que se están optimizando.

² La distancia métrica de Minkowski define una familia distancias [27]. Se le conocen también como métricas o normas L_p y esta dada por:

$$d_{ij} = \left[\sum_{l=1}^t |x_{il} - x_{jl}|^p \right]^{1/p}$$

donde p es un número real positivo tal que $p \geq 1$.

de n variables independientes \mathbf{v} ³. En nuestro caso el número de variables es grande porque usamos la frecuencia de cada intensidad de color en el espacio RGB para formar el espectro de la imagen. Cada intensidad de color corresponde a una variable y hay 256 de ellas. Para atacar este problema analizamos dos enfoques de Aproximación Multivariada. El primero se basa en consideraciones algebraicas, utilizando un aproximador de tipo polinomial. Este enfoque es el más adecuado porque la relación entre las variables del modelo es explícita y nos permitiría analizar las interrelaciones entre ellas. La primera decisión que debemos tomar es la referente a la forma del aproximador. Por simplicidad seleccionamos aproximadores polinomiales. Sabemos, del teorema de Weirstrauss⁴, que tienen una capacidad expresiva universal para funciones continuas. Utilizando un polinomio de colocación nos aseguramos que todos los datos estén en el modelo. El número de coeficientes del polinomio de colocación, dado el número de variables, por tanto, requerirá de, al menos 256 valores. Sin embargo, al hacer esto se nos presentan dos inconvenientes: a) Perdemos la capacidad de generalización del modelo. b) Requerimos de una mayor cantidad de datos para que encontrar el modelo $f(\mathbf{v})$ que represente correctamente el sistema. Así pues, se nos presentan 2 problemas: 1) la alta dimensionalidad⁵ y 2) la cantidad de memoria requerida para resolver los coeficientes del modelo. Por ejemplo, para encontrar los coeficientes requerimos resolver sistemas de ecuaciones como (1). En donde m es el número de coeficientes, y X_{mm} . Para el número variables considerado, tendríamos pues, que resolver sistemas de ecuaciones de tamaño de por lo menos 256×256 .

$$\begin{aligned} \varepsilon_1 + c_1 X_{11} + c_2 X_{12} + \cdots + c_m X_{1m} &= f_1 \\ \varepsilon_2 + c_1 X_{21} + c_2 X_{22} + \cdots + c_m X_{2m} &= f_2 \\ \vdots & \\ \varepsilon_M + c_1 X_{m1} + c_2 X_{m2} + \cdots + c_m X_{mm} &= f_m \end{aligned} \quad (1)$$

El número de variables puede ser menor si estas se eligen selectivamente, por ejemplo utilizando análisis de componentes principales (PCA). Sin embargo, en nuestro caso, deseamos obtener un modelo de aproximación y no de colocación. Nuestra tesis es que un modelo aproximado, con datos elegidos adecuadamente, engloba el comportamiento de todas las observaciones. Es decir, podemos usar más de los 256 valores mínimos pero demandando que el número de coeficientes sea razonablemente pequeño.

Examinamos 3 diferentes algoritmos para determinar cuál de estos tres se desempeña mejor y decidir cuál utilizar preferentemente: a) el Algoritmo de Levenberg-Marquardt (LM) [38, 49], b) el Algoritmo de Powell (PM) [47] y (c) el Algoritmo de Ascenso (Fast Ascent Algorithm FA) [8]. Tanto (a) como (b) están basados en una combinación del método del gradiente descendente y un método basado en la serie de Taylor. Ambos minimizan la norma L_2 . FA es un algoritmo de intercambio y minimiza la norma L_∞ . El resultado de nuestros análisis muestra, por un lado, que tanto (a) como (b) requieren de grandes cantidades de memoria para resolver un sistema de ecuaciones y encontrar la solución. Incluso con una cantidad relativamente limitada de variables el sistema de ecuaciones en (a) tiende a ser inestable, no así en (b). Por otro lado (c), al ser un algoritmo de intercambio, no requiere de

³ Es un vector de variables independientes. $\mathbf{v} = (v_1, v_2, \dots, v_n)$

⁴ Teorema de aproximación de Weirstrauss, Cap 3.3 pag 66 en [8].

⁵ Una función definida en un espacio de alta dimensiones tiene una alta probabilidad de ser más compleja que una definida en un espacio de bajas dimensiones. Mientras aumenta la dimensionalidad del problema, la cantidad de datos necesarios requeridos aumenta exponencialmente [9].

gran cantidad de memoria ya que no necesita resolver grandes sistemas de ecuaciones. Esta característica lo hace más útil que los otros dos para efectos prácticos. El problema de la multi-dimensionalidad expuesto anteriormente lo atacamos incorporando a (a), (b) y (c) un Algoritmo Genético para seleccionar los términos más significativos del aproximador y de esta forma crear lo que llamamos Polinomios Genéticos Multivariados (GMP) [30, 36]. El segundo enfoque son las Redes de Perceptrones Multicapa (MLPs). Se siguió una metodología para determinar la mejor arquitectura de la red [28] tomando en cuenta el número de variables y la cantidad de información en los datos.

Una parte fundamental del análisis consistió en encontrar la mejor forma de estructurar la información para lograr una buena tasa de aprendizaje. Encontramos que la mejor forma es expresar los datos como las frecuencias de los colores (RGB) de los puntos encontrados por el método del Cedazo. Esta organización de los datos nos permite identificar las imágenes hasta para una gran cantidad de variables logrando un porcentaje de reconocimiento del 100% en una base de datos experimental de muestras que consta de 5 imágenes. En trabajos posteriores se buscará generalizar este resultado.

Por otro lado, durante el desarrollo de este trabajo se determinó que un punto vital para el buen funcionamiento de las técnicas utilizadas es necesario realizar un acondicionamiento previo de los datos. El acondicionamiento o pre-procesamiento consiste en: 1) escalar los datos en un rango predeterminado, 2) estabilizar el sistema agregando una perturbación mínima a los datos $O(10^{-6})$, 3) realizar un análisis de correlaciones, 4) mejorar los datos utilizando splines naturales y 5) calcular la cantidad de información en los datos.

En el enfoque que estamos planteando esta tácito hay mucha información que queremos que sea procesada en tiempo real. Por lo tanto nuestro problema fundamental consiste en manipular dicha información de manera tal que sea muy eficiente. Por lo tanto en resumen, 1) se resolvió un problema multi-objetivo para optimizar el proceso “on-line” que permite ubicar el robot. 2) Se analizó la posible utilización de tres algoritmos evolutivos multi-objetivo. 3) De acuerdo con la metodología adoptada se pre-procesaron los datos para cada uno de los algoritmos. 4) Se examinaron diferentes formas de estructurar los datos. 5) Se exploraron dos enfoques de aprendizaje supervisado. a) Métodos de Aproximación Multivariada basados en consideraciones algebraicas: i) LM, ii) PM y iii) FA. En ellos se manifiesta el problema de la alta dimensionalidad de los coeficientes. 6) Este problema se resolvió utilizando Polinomios Genéticos Multivariados. Dado que los GMP son muy sensibles al número de variables, exploramos b) las MLPs. 7) Para ellas se determinó la mejor arquitectura. Con esta logramos una tasa de aprendizaje del 100% para la base de imágenes considerada.

2 Objetivo

Implementar un método de localización para un robot móvil.

Para cumplir este objetivo es necesario:

- a) Identificar la información mas relevante de un conjunto de imágenes para obtener la firma espectral de cada una de ellas.
- b) Establecer un método para la identificación de las firmas espectrales obtenidas de las imágenes.

El método que proponemos no provee un mapa del entorno, ni realiza un emparejamiento de características de forma probabilística. Sino busca la posición del robot utilizando el espectro de las imágenes que están formadas en base a una colección puntos.

3 Antecedentes

Varios métodos han sido propuestos para identificar la posición de un robot móvil en un ambiente conocido [10, 18, 24, 56]. Estos métodos se valen de distintos tipos de sensores (infrarrojos, ultrasónicos, laser, cámaras, etc.), para obtener una gran variedad de mediciones (distancia, temperaturas, marcas en el ambiente, etc.). Su objetivo es facilitar al robot la posibilidad de identificar su posición. Sin embargo, no siempre es posible incorporar marcas en el ambiente o utilizar los sensores antes mencionados. Esto se debe a que la topografía del ambiente hace difícil incorporar las marcas o el costo de los sensores es muy elevado o por especificaciones del problema a resolver.

En una etapa previa (Maestría) se introdujo el método del Cedazo (sieve) [31, 33, 42] para determinar la posición de un robot a partir de un conjunto de imágenes utilizando estrategias de aprendizaje de máquina. El método determina la posición al comparar una imagen adquirida durante su recorrido contra un conjunto de imágenes del ambiente tomadas previamente. A diferencia de otros métodos, con este no se requiere incorporar marcas en el ambiente para identificar una imagen o la posición del robot. El método del Cedazo determina el número mínimo de píxeles necesarios y su posición en la imagen. Esto hace que la comparación de imágenes e identificación de la posición se realice eficientemente. Al determinar el número mínimo necesario de píxeles se puede llegar a reducir el tamaño de los datos que se tienen que analizar hasta en un 99%.

Durante el proceso de investigación para perfeccionar el método nos encontramos con el método de Villegas-Quezada [60] que consiste en el reconocimiento holístico de rostros, el cual utiliza lo que llama el espectro característico del rostro humano. A diferencia de otros métodos, no se basa en características aisladas ni elementos geométricos en el rostro. En su lugar propone un “espectro” compuesto por distintos valores como por ejemplo: frecuencia, gradiente y entropía de la imagen de un rostro. Sin embargo, los valores que componen el espectro del rostro pueden ser reemplazos o complementados con otros. Para identificar un rostro, primero se busca un polinomio característico para cada uno de los valores que conforman el espectro de los rostros a reconocer por medio de una aproximación multivariada utilizando algoritmos genéticos⁶. El reconocimiento de un rostro se logra comparando el espectro del rostro a identificar contra las aproximaciones polinomiales del conjunto de entrenamiento utilizando como métrica la distancia de Mahalanobis⁷. Dado el método ante-

⁶ Los Algoritmos Genéticos fueron introducidos originalmente por Holland como una abstracción del proceso biológico de evolución [51]. Son una familia de métodos computacionales utilizados para resolver problemas de optimización y satisfacen las siguientes condiciones [29]:

1. Trabajan en un espacio discreto n-dimensional.
2. Analizan simultáneamente un conjunto finito de posibles soluciones.
3. Los elementos que determinan las soluciones llamados individuos, están codificados explícitamente.
4. Los individuos se evalúan mediante una función de aptitud.
5. Se seleccionan los mejores individuos para mejorar la búsqueda en el espacio de posibles soluciones.
6. Porciones del código de los individuos se combinan periódicamente.
7. Porciones del código de los individuos se modifican periódicamente de forma aleatoria.
8. Se mantiene un subconjunto de los mejores individuos.
9. El algoritmo repite los pasos 4-8 hasta que se cumple una condición de paro.

⁷ La distancia de Mahalanobis [48] es la distancia entre un punto de los datos y el centroide de un espacio multivariado (media general). Esta distancia toma en cuenta las diferentes escalas entre las variables y las correlaciones entre éstas.

$$\text{Se define como } D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$$

rior nuestra tesis es que es factible aplicarlo al reconocimiento de cualquier otro tipo de objetos o lugares. De igual forma es posible usar distintos valores para crear los “espectros” de las imágenes, además de que es posible utilizar diferentes métricas para la identificación.

4 Análisis del estado del arte

El problema que queremos resolver es identificar la posición del robot en un ambiente conocido. Nuestra forma de resolver este problema es distinta a lo que hemos visto en la literatura. En lo que sigue haremos un breve repaso del estado del arte de las técnicas conocidas y al final hablaremos de nuestro método.

Se entiende por navegación robótica la acción de ir de un punto de origen a un punto destino [10]. Es por eso que cobra una gran importancia que un robot pueda determinar de alguna forma su posición.

De acuerdo con diferentes estudios existen dos grandes enfoques [3, 18] en la localización y navegación utilizando visión en robots móviles dentro de ambientes internos:

1. Navegación basada en mapas
2. Navegación sin mapa

Este problema ha sido abordado utilizando diferentes tipos y combinaciones de sensores, lo cual ha llevado a un gran número de soluciones [3, 4, 18].

En la navegación basada en mapas los robots necesitan tener un conocimiento previo de todo el ambiente, i.e. un mapa o modelo con diferentes niveles de detalle para que el robot pueda navegar. Dicho modelo puede ser proporcionado al robot previamente o puede ser generado por el mismo para ocuparlo en una etapa posterior. De esta forma cuando el robot recorre el ambiente, busca marcas que puede identificar y comparar contra las que tiene almacenadas en el mapa. Los principales pasos en la navegación por mapas son [56]:

1. Percepción: Se adquieren datos de los sensores para obtener información. e.g. Imágenes de una cámara, lecturas de distancia, etc.
2. Localización: El robot determina su posición en el ambiente al identificar marcas u objetos.
3. Planeación de movimientos: El robot determina que acciones debe ejecutar para llegar a su objetivo.
4. Ejecución: El robot ejecuta las acciones y actualiza su posición en función de las marcas identificadas en el mapa.

Dentro de estos métodos por un lado se encuentran estrategias de localización absoluta y las estrategias de localización relativa [18]. En la localización absoluta se desconoce la posición inicial del robot y se usan algoritmos para que el robot se auto-localice, tales como localización de Markov [59] o Montecarlo [17]. En la localización relativa se asume que se conoce la posición en la que el robot inicia su navegación y esta se actualiza incrementalmente mientras se va moviendo por el ambiente. En este tipo de localización se define un umbral de error en la aproximación de la posición, cuando se sobrepasa dicho umbral el robot utiliza los sensores para determinar la nueva posición. En estas estrategias se opta por técnicas para reducir los errores certidumbre tales como el uso de cámaras estereoscópicas [50].

donde $\vec{x} = (x_1, x_2, \dots, x_N)^T$ con media $\vec{\mu} = (\mu_1, \mu_2, \dots, \mu_N)^T$ y una matriz de covarianza S .

En la navegación sin mapas, el robot descubre el ambiente mientras navega por él. En los métodos basados en este enfoque el robot no tiene conocimiento previo del ambiente. Generalmente posee comandos de control que simulan comportamientos de tipo reactivo y que dependen de marcas o características que el robot vaya identificando en el ambiente [3]. Por lo tanto no posee una representación global del mismo. En la navegación sin mapa se ocupan técnicas tales como flujo-óptico (optical-flow) [6] y las llamadas técnicas de emparejamiento por aspecto (appearance-based matching techniques) [3]. En las técnicas de flujo-óptico se estima el movimiento de los objetos en una secuencia de imágenes. Siendo los pioneros en estas técnicas son Horn [21] y Lucas-Kanade [45]. Estas técnicas pueden aprovechar mejoras tales como extraer previamente de las imágenes atributos predominantes tales como bordes o esquinas [26], o combinar el uso de cámaras estereoscópicas para estimar la profundidad y detectar objetos en movimiento en la escena [57].

Las técnicas de emparejamiento por aspecto constan de dos fases [3]: 1) la fase de pre-entrenamiento y 2) la fase de navegación. En 1) se almacenan moldes que pueden consistir en un conjunto de imágenes o ser las características más importantes del ambiente. Los moldes se asocian con información referente a la localización y/o con un comando de control. En 2) el robot se auto-localiza relacionando las imágenes que capta con los moldes almacenados. El reto más importante en estas estrategias es encontrar un algoritmo que: a) genere una representación del ambiente y b) defina el criterio de identificación en la fase de navegación. Por ejemplo Zho [62] aborda este problema utilizando una base de datos de histogramas multidimensionales en lugar de imágenes. El reconocimiento del ambiente en la fase de navegación se realiza comparando el histograma multidimensional de la imagen captada contra el histograma que tiene almacenado.

Tanto el método que proponemos, utilizando el espectro de las imágenes, así como la versión original del método del Cedazo no se caen en ninguna de las clasificaciones antes mencionadas. En nuestra propuesta tenemos *a priori* un modelo global del ambiente que se quiere navegar que consiste en un conjunto de imágenes independientes. Sin embargo no se provee un mapa del entorno, ni se descubre el ambiente, ni se identifican cosas o marcas de manera probabilística. La posición del robot se determina a partir de una colección puntos determinados a partir de dicho conjunto de imágenes. Aquí el problema es que existe mucha información y tenemos que determinar, de alguna manera, la más relevante. De esta forma lo que buscamos es quitar la incertidumbre presente en las técnicas tradicionales de SLAM. Este enfoque, hasta donde sabemos, es original.

5 Metas

Por lo expuesto ya anteriormente la implementación del método propuesto en el presente trabajo busca reducir la cantidad de información presente en el modelo del ambiente. Al igual que en las técnicas de “emparejamiento por aspecto”, consta de dos etapas a) pre-entrenamiento y b) navegación. En estas están presentes las siguientes metas:

- Para la fase de pre-entrenamiento, necesitamos: a) encontrar una forma alterna para extraer la información más relevante del conjunto de imágenes que definen el ambiente a navegar. b) La información extraída será identificada como la firma espectral de la imagen. El método del Cedazo utiliza un método de optimización multiobjetivo para encontrar el conjunto óptimo de píxeles característicos. Analizaremos si este conjunto óptimo de píxeles puede ser utilizado para encontrar la firma espectral. Con la firma espectral de la imagen se busca reducir el tamaño de la informa-

ción y representar ambientes complejos manteniendo el proceso de identificación de los mismos simple.

- Para la fase de navegación, se usarán técnicas de aprendizaje de máquina tales como: a) Polinomios Multivariados [30, 41] y/o b) Redes de Perceptrones [20]. En el caso de (a) la firma espectral de las imágenes estará representada por uno o varios polinomios. En (b) estará representada por modelos conexionistas. El robot móvil utilizará las representaciones del espectro de la imagen para identificar su posición en un momento dado. Nótese que esta identificación es independiente del punto de salida del robot.

6 Metodología

A continuación describiré las técnicas utilizadas durante el desarrollo de la investigación. Para la fase de pre-entrenamiento, se exploraron 2 métodos para extraer la información y crear la firma espectral. Estos métodos son: 1) El método del Cedazo y 2) el algoritmo ORB (Oriented FAST and Rotated BRIEF) [36].

6.1 Optimización multi-objetivo

El primer problema que vamos a atacar es determinar la información mínima requerida para identificar una imagen con un error pequeño y con esta información crear el espectro de las imágenes. Para resolverlo utilizaremos optimización multi-objetivo. Antes de pasar a detallar cómo resolvemos el problema definiremos en qué consiste un problema multi-objetivo. A los problemas de optimización en donde se requiere satisfacer simultáneamente varios objetivos se les conoce como problemas de optimización multi-objetivo. Formalmente, un problema multi-objetivo consiste en encontrar un conjunto de n variables de decisión que optimicen un conjunto de M funciones objetivo que pueden estar sujetas a K restricciones de igualdad y J restricciones de desigualdad [52]. El modelo del problema es el siguiente:

$$\begin{aligned} \text{Optimizar} \quad & f_m(x_i), \forall m = 1, 2, \dots, M \\ \text{sujeto a:} \quad & g_j(x_i) \leq 0, \forall j = 1, 2, \dots, J \\ & h_k(x_i) = 0, \forall k = 1, 2, \dots, K \end{aligned} \quad (2)$$

Donde $x_i = \{x_1, x_2, \dots, x_n\}$ es el vector de n variables de decisión.

Este tipo de problemas se abordan utilizando técnicas evolutivas dando origen al término Algoritmos Evolutivos de Optimización Multi-objetivo (MOEAs). En general existen 2 enfoques [13] para atacar estos tipos de problemas: a) Combinar las funciones objetivos en una sola función ú b) optimizar simultáneamente cada una de ellas. Los algoritmos basados en el enfoque (a) son llamados “a priori” ya que de antemano se establece un compromiso o preferencia entre las funciones a optimizar asignando a cada una un peso de acuerdo a estas preferencias. La asignación de los pesos se efectúa antes de ejecutar la optimización y no puede ser modificada durante la búsqueda de la solución. Establecer los compromisos entre las funciones objetivo en estos algoritmos requiere de un profundo conocimiento del problema a optimizar. Por lo que es necesario crear un modelo del problema lo cual, generalmente, no es posible. Además no es recomendable ya que puede introducir relaciones incongruentes entre estas funciones. Los algoritmos basados en (b) son llamados “a posteriori”. No encuentran una solución única sino un conjunto de soluciones potenciales tales que

las componentes del vector objetivo no pueden necesariamente mejorar simultáneamente. A este conjunto de soluciones se le conoce como el conjunto de soluciones óptimas de Pareto⁸. En una etapa posterior, un experto elige la solución basado en sus preferencias. Métodos cómo RPSGAe (Reduced Pareto Set Genetic Algorithm with elitism) [19], NSGA-II (Non-dominated Sorting Genetic Algorithm) [16] y MO-SHERPA (Multi-Objective SHERPA) [7] son algunos ejemplos de MOEAs. Además de los algoritmos “a priori” y “a posteriori”, también existen algoritmos llamados “interactivos” que permiten alterar sus parámetros durante la ejecución para poder guiar la dirección de la búsqueda de la solución. Sin embargo estos algoritmos requieren que el experto se encuentre disponible en todo momento para realizar estos cambios. En general estos métodos tienen un alto costo computacional, debido que exploran simultáneamente diferentes puntos en el espacio solución para cada una de las funciones objetivo a optimizar. Mientras mas grande sea el número de funciones objetivo a optimizar el costo computacional es mayor disminuyendo así su desempeño. Esto se ve reflejado en estudios experimentales [7, 63] en donde los métodos se validan utilizando conjuntos de funciones en las que hay un máximo de dos funciones objetivo a optimizar.

6.2 Método del Cedazo (Sieve Method)

Introdujimos el método del Cedazo [31, 33] con el propósito de reducir la cantidad de información a analizar en una imagen. Para aplicar el método es necesario obtener un modelo del mundo (M) que consiste de una secuencia de imágenes (I) que incluyen a cada uno de los posibles cuadros visuales (V) del robot en cada una de las posibles ubicaciones (U) dentro del ambiente. Típicamente el número de posibles escenarios en (V) es demasiado grande por lo que es necesario acotar tanto las posiciones del robot en (U) como (V). Las imágenes están formadas por P píxeles. En total, pues, M consta de $T = P \times U$ píxeles. Para que el robot (R) pueda identificar su posición al tiempo t , R debe capturar $V(t)$ y compararlos con las imágenes en U . El número de comparaciones a realizar es elevado aún para M 's de tamaños pequeños. Sin embargo, no es necesario comparar cada uno de los píxeles de los V cuadros. Es posible determinar un subconjunto de T tal que el número de aciertos (X) en cada comparación se maximice al mismo tiempo que el número de píxeles a comparar (Y) sea mínimo. Encontrar los valores óptimos de X y Y constituye un problema de optimización multi-objetivo. Donde X y Y pueden estar en conflicto, ya que por un lado buscamos el conjunto de píxeles más grande posible para que la identificación sea precisa. Y por otro lado, deseamos que sea lo más pequeño posible para que la identificación sea veloz. Para encontrar la solución a este problema se eligió el algoritmo RPSGAe (Reduced Pareto Set Genetic Algorithm with elitism) cuyo pseudocódigo se muestra a continuación.

Algoritmo RPSGAe:

- 1) Generar la población inicial (población interna)
 - 2) Crear una población externa
- While** not Stop-Condition **Do**
- a) Evaluar la población interna
 - b) Calcular el fitness de los individuos usando una técnica de cluster
 - c) Ordenar la población respecto a su fitness
 - d) Copiar los mejores individuos a la población externa

⁸ Las soluciones óptimas de Pareto consisten en un conjunto de soluciones que son consideradas como buenas y que ninguna es mejor que las otras. También se les conoce como el conjunto de soluciones no dominadas [52].

```

If población externa está llena
    Copiar los n mejores individuos a la nueva población
End If
    e) Seleccionar los individuos para generar la nueva población
    f) Aplicar operador de cruzamiento.
    g) Aplicar operador de mutación.
End While
□

```

El algoritmo trabaja sobre una población de individuos codificados como una cadena de bits de tamaño igual al número de píxeles en una imagen. Cada individuo es una configuración de un cedazo donde un elemento en la cadena corresponde a la posición de un píxel en la imagen. Un elemento encendido corresponde a un píxel que forma parte del cedazo de forma que cada individuo puede tener un cedazo de tamaño diferente. En la figura 1 se muestra un ejemplo de 10 individuos diferentes. Un cuadro negro representa un píxel seleccionado que forma parte del cedazo para ese individuo.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	SIZE
1					■																													5		
2						■																												6		
3							■																											4		
4	■							■																										6		
5									■																									9		
6										■																								4		
7		■									■																							5		
8											■																							7		
9												■																						9		
10													■																					4		

Figura 1. Genoma de 10 individuos para imágenes de 35 píxeles.

Los operadores genéticos utilizados en el algoritmo están basados en los usados en el Algoritmo Genético Ecléctico (EGA) [37], el cual ha mostrado ser el mejor de entre una familia de algoritmos genéticos estructuralmente diferentes [29, 35]. Estos operadores se muestran en la figura 2 y son: 1) la selección, la cual se realiza de forma determinística tomando el mejor y el peor de los individuos para su cruzamiento. 2) El cruzamiento se realiza de forma anular. Se toman al azar 2 individuos y se seleccionan aleatoriamente 2 puntos en el código del individuo. Las partes del código genético definidas por estos puntos se intercambian entre los individuos seleccionados. 3) La mutación se realiza de manera uniforme. Se calcula el número total de elementos del código a mutar a priori y se mutan seleccionando de manera aleatoria los individuos y la posición de los elementos en el código a mutar. Los individuos se evalúan contando el número de píxeles hay en el cedazo y cuántas veces aparece el mismo valor del píxel en esa posición en todas las imágenes.

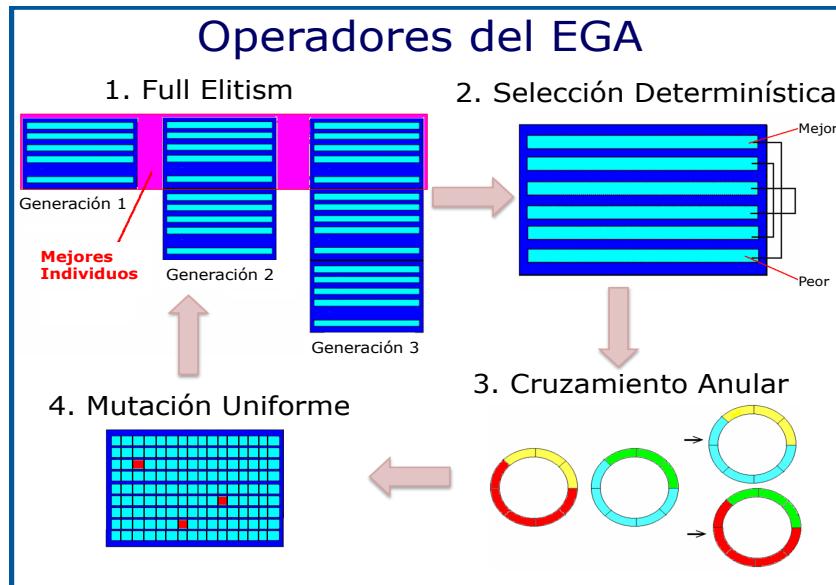


Figura 2. Operadores del EGA.

El método del Cedazo original arroja como resultado un vector de posibles soluciones, figura 3, de las cuales seleccionamos el punto medio de este vector para determinar el número de píxeles que conforman el cedazo.

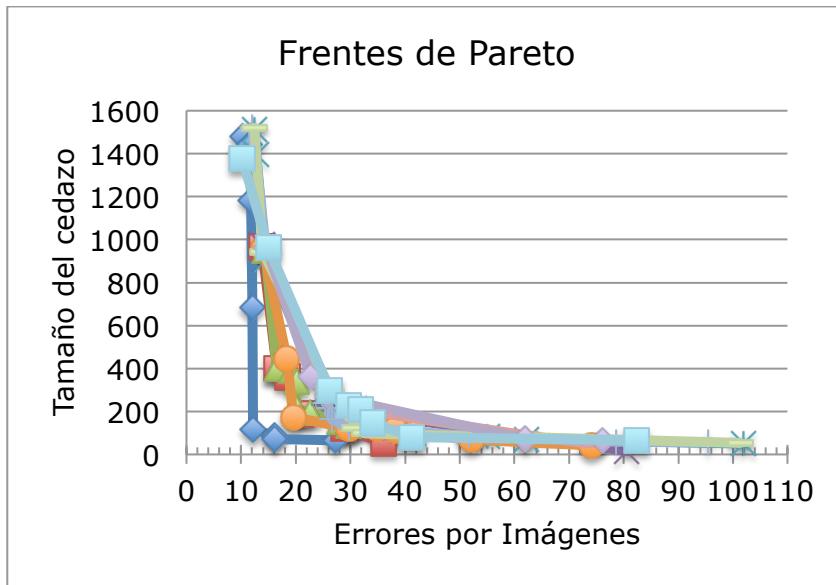


Figura 3. Frentes de Pareto obtenidos en el método del Cedazo.

6.3 Optimización multi-objetivo basado en la norma L_∞

Los MOEA, al ser algoritmos iterativos que resuelven simultáneamente un vector de funciones objetivos al buscar soluciones candidato en un espacio de solución grande, son computacionalmente intensivos. Esto tiene un impacto considerable al trabajar con grandes cantidades de información, como es el caso con el número de píxeles en una imagen. Para solucionar este problema abordamos nuevamente el problema multi-objetivo en el método del Cedazo utilizando un enfoque basado en la norma L_∞ . Este enfoque ya había sido pro-

puesto anteriormente [12] y limita el espacio de búsqueda de la solución al seleccionar “a priori” los criterios de optimización. Con la norma L_∞ estamos seleccionando “a priori” la decisión de minimizar el peor-de-los-casos posibles que se presente al evaluar simultáneamente las funciones objetivo. El criterio de dominancia genérica se selecciona “a priori” y obtenemos los valores que minimizan este criterio. En este caso esta estrategia garantiza que todas las variables del problema no sean dominadas más que por una de ellas. En ese sentido esto equivale a aceptar que una de dichas variables (aquella cuyo valor máximo determina los mínimos de las otras) domine a todas las demás y nos lleva a la elección de los valores que satisfacen el criterio de dominancia. La solución entregada por el algoritmo consiste en un solo punto.

Realizamos una validación experimental del algoritmo para comparar nuestro resultados por los reportados en [7]. Para lo cual resolvimos un conjunto de funciones diseñadas específicamente para probar algoritmos multi-objetivo [63] mismas que fueron probadas en el trabajo ya citado y se muestran en la tabla 1. Cada función se ejecuta 25 veces utilizando diferentes combinaciones de tamaño la población, las cuales tomaron valores de 12, 40 y 100 individuos, y de generaciones, las cuales tomaron valores de 500, 1,000, 2,000, 5,000, 7,500, 10,000 12,500, 20,000 y 25,000 dependiendo el problema. La probabilidad de cruce $P_c = 1.0$ y la probabilidad de mutación $P_m = 0.01$ permanecieron fijas en todo momento. Los resultados que obtuvimos muestran que en cada una de las funciones la mejor solución entregada por el algoritmo forma parte del conjunto de soluciones óptimas de Pareto sin haber calculado todas las posibles soluciones de manera explícita. Esto se muestra en las gráficas de la figura 4.

Tabla 1. Funciones multi-objetivo utilizadas en la validación experimental.

Nombre	Función	Notas
ZDT1	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1)$ $h(f_1, g) = 1 - \sqrt{f_1/g}$	Frente de Pareto convexo. Se obtiene cuando: $g(x) = 1.0$ donde: $m = 30$ $x_i \in [0,1]$
ZDT2	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1)$ $h(f_1, g) = 1 - (f_1/g)^2$	Tiene un frente de Pareto no-convexo que se obtiene cuando: $g(x) = 1.0$ donde: $m = 30$ $x_i \in [0,1]$
ZDT3	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \sum_{i=2}^m x_i / (m-1)$ $h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g) \sin(10\pi f_1)$	Incluye una característica discontinua introducida por la función seno, el frente de Pareto consiste en varias secciones no-convexas que aparecen cuando $g(x) = 1.0$. donde $m = 30$ and $x_i \in [0,1]$.

Nombre	Función	Notas
ZDT4	$f_1(x_1) = x_1$ $g(x_2, \dots, x_m) = 1 + 10(m-1) + \sum_{i=2}^m (x_i^2 - 10 \cos(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{f_1/g}$	<p>Esta función contiene 21^9 frentes de Pareto. Esta diseñada para tratar la multimodalidad del algoritmo.</p> <p>Donde :</p> <p>$m = 10$ and $x_1 \in [0,1]$ and $x_2, \dots, x_m \in [-5,5]$</p> <p>El frente de Pareto global aparece con $g(x) = 1.0$. El mejor frente de Pareto aparece con $g(x) = 1.25$</p>
ZDT6	$f_1(x_1) = x_1 - \exp(-4x_1) \sin^6(6\pi x_1)$ $g(x_2, \dots, x_m) = 1 + 9 \cdot \left(\left(\sum_{i=2}^m x_i \right) / (m-1) \right)^{0.25}$ $h(f_1, g) = 1 - (f_1/g)^2$	<p>La función no tiene un espacio de búsqueda uniforme. Las soluciones de Pareto no se distribuyen uniformemente.</p> <p>El frente de Pareto no es convexo y aparece con $g(x) = 1.0$</p> <p>Donde $m = 10$ and $x_i \in [0,1]$</p>

En las gráficas (a) a la (f) de la figura 4 se muestra el frente de Pareto asociado al problema que se está resolviendo, el número de individuos, el número de generaciones y las veces que se ejecutó el problema así como los puntos de la solución encontrados en cada una de las corridas del algoritmo basado en L_∞ . Comparando nuestros resultados con los reportados en [7], en todas las funciones logramos obtener puntos solución que pertenece al frente de Pareto con pocos individuos y relativamente un número de generaciones menor. El detalle de los resultados para cada problema se muestra en el anexo 1.

Es importante a resaltar que para hacer una comparación directa entre las funciones a optimizar es fundamental que estas se encuentren en la misma escala de magnitud. Por lo que el escalamiento es un paso fundamental antes de evaluar las funciones.

Se puede hacer una comparación estimada de la velocidad en la que se encuentra una solución entre los enfoques tradicionales y el nuestro. En general los MOEA basados en encontrar el frente de Pareto son un orden de magnitud mayores. Dependiendo del algoritmo que se use en cada iteración evalúan la población para determinar: a) el clúster al que pertenece cada individuo o b) la calificación de los individuos para hacer que se distribuyan uniformemente. En el enfoque que proponemos no pasa esto. Por lo tanto una de las ventajas de nuestro enfoque es que el costo computacional no aumenta, incluso si el número de funciones objetivo a optimizar se incrementa.

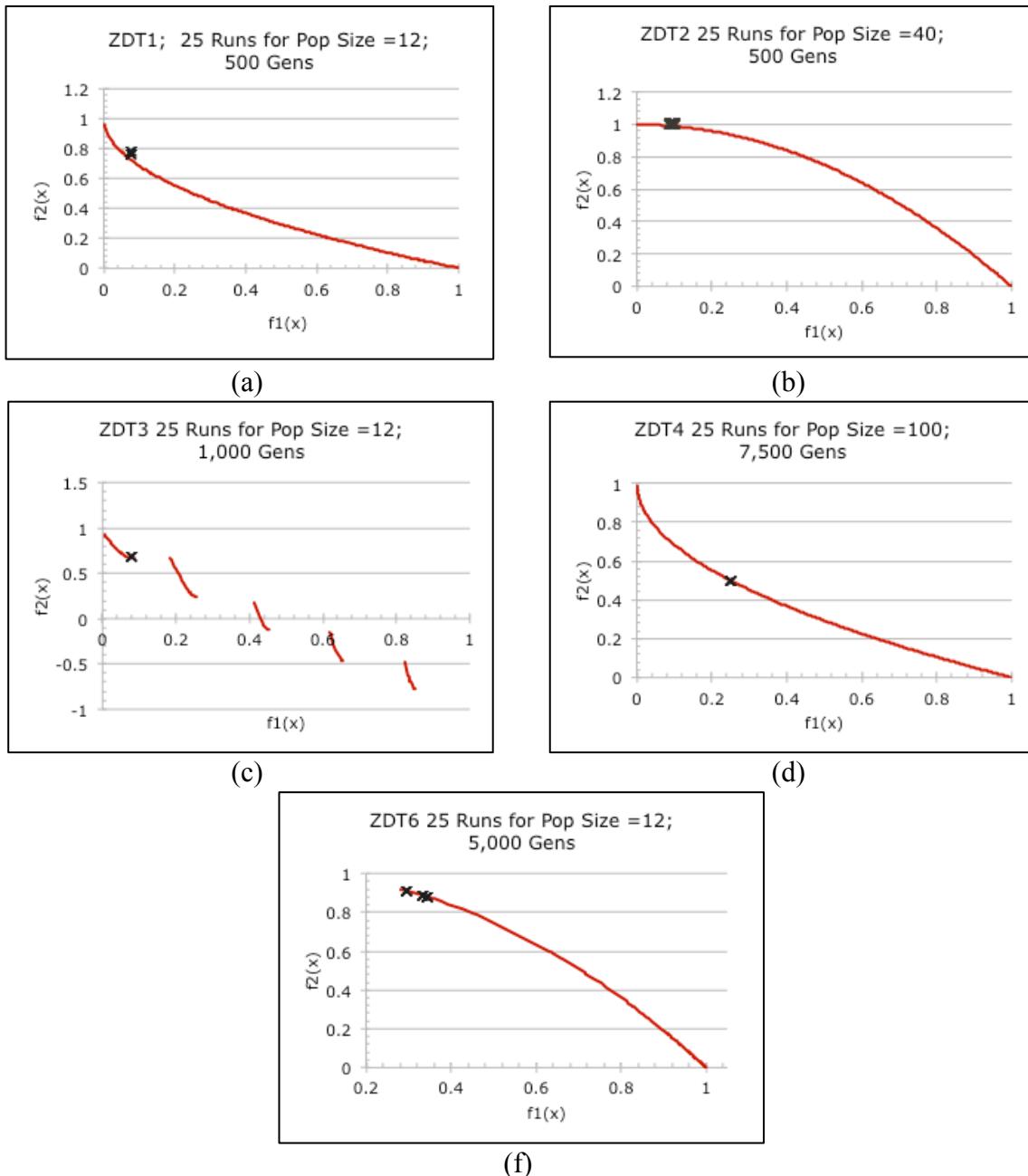


Figura 4. Frentes de Pareto (en rojo) de los 5 problemas en los que hicimos la prueba. Los puntos que encontró MOEGA caen sobre el frente de Pareto.

Las conclusiones de este estudio nos dieron certidumbre para utilizar este enfoque. Además de que el problema que queremos resolver en el método del Cedazo este enfoque resulta el adecuado por lo ya expuesto. Al nuevo algoritmo multi-objetivo con la norma L_∞ lo llamamos MOEGA (Multi-Objective Eclectic Genetic Algorithm), y su pseudocódigo es el siguiente.

Algoritmo MOEGA:

1. Generar una población inicial.
2. Evaluar la población en ambas funciones objetivo.

While not Stop-Condition **Do**

- a) Escalar los valores obtenidos de las funciones y se comparan usando la norma L_∞ . $\text{fitness} = \max(f_1, f_2, \dots, f_n)$
- b) Ordenar los individuos de menor mayor. Se minimiza el máximo de los errores.
- c) Duplicar la población para un elitismo completo.
- d) Hacer un acoplamiento determinístico con los primeros n individuos. Se toma el mejor individuo con el peor individuo
- e) Aplicar operador de cruzamiento (cruzamiento anular)
- f) Aplicar operador de mutación (mutación uniforme)
- g) Evaluar la población en ambas funciones objetivo.

End while

□

Después de incorporar el MOEGA en el método del Cedazo, comparamos las dos versiones del método identificando imágenes utilizando los puntos del cedazo encontrados por los dos algoritmos [32].

Con la versión original del método del Cedazo, que utiliza el algoritmo RPSGAe, encontramos una configuración 12 puntos con un error de 2.4736, mientras que con la versión que utiliza la norma L_∞ obtuvimos una configuración de 31 puntos y un error de 0.000269. El reconocimiento de las imágenes lo realizamos calculando la distancia promedio entre los píxeles del cedazo de la imagen capturada contra cada una de las imágenes que conformaban la base de datos como se describe en [43]. La imagen de referencia, cuya distancia es la menor, indica que es la que más se parece a la imagen que deseamos identificar. En la figura 5 se muestran las imágenes que deseamos reconocer y las imágenes identificadas utilizando puntos con ambas versiones del método del Cedazo. La tasa del reconocimiento utilizando 12 puntos obtenidos por la versión original del método del Cedazo fue del 67% mientras que utilizando los 31 puntos obtenidos por la versión basada en la normal L_∞ fue del 79%.

Imagen tomada de una posición conocida	Imagen identificada en la base de datos usando 12 píxeles	Imagen identificada en la base de datos usando 31 píxeles
 4x, 3y, 60 θ	 5x, 4y, 60 θ	 4x, 3y, 60 θ
 2x, 5y, 300θ	 3x, 0y, 210 θ	 2x, 5y, 300θ

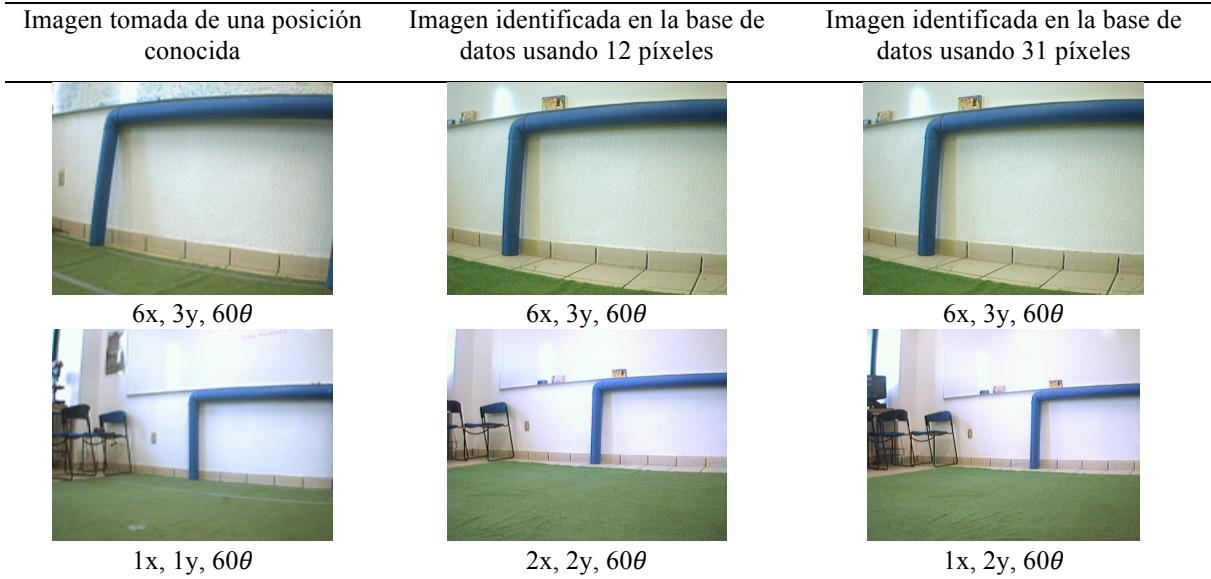


Figura 5. Imágenes identificadas utilizando los puntos obtenidos con dos versiones del método del Cedazo. Usando RPSGAe y la norma L^∞ .

6.4 ORB

El método del Cedazo nos permite definir el espectro de la imagen con los puntos obtenidos con el fin de lograr una mejor representación del ambiente. La firma espectral nos debe ayudar a reducir la cardinalidad del conjunto preservando la información. Como ya lo habíamos mencionado anteriormente, dado que la firma espectral puede estar representada por diferentes valores, además de los obtenidos por el método del Cedazo, analizamos el algoritmo ORB (Oriented FAST and Rotated BRIEF) [54] como otra opción para obtener los valores que compongan la firma espectral. ORB es una alternativa a los algoritmos SIFT [44] y SURF [1]. Utiliza una versión modificada del detector de puntos característicos FAST [53] y del descriptor BRIEF [5]. Con FAST se obtienen puntos característicos que son corresponden a esquinas. Estos puntos pueden ser multi-escala e incorporan una componente de rotación (el ángulo) de los puntos detectados. Los descriptores de los puntos característicos se obtienen con rBRIEF⁹. Los descriptores encontrados consisten en una cadena de bits que se obtiene a partir de un conjunto de pruebas de intensidad. Las pruebas de intensidad son aplicadas en dos puntos aleatorios en una región de la imagen centrada en cada uno de los puntos característicos. Para identificar descriptores entre dos imágenes, ORB utiliza LSH multi-sonda (Multi-Probe Local Sensitivity Hashing) [46]. Los puntos característicos y los descriptores obtenidos por este algoritmo pueden ser considerados también para formar la firma espectral de las imágenes.

Ambos algoritmos obtienen los puntos de forma diferente. Aún no hemos realizado un análisis cuantitativo para comparar ambos métodos. Por un lado creemos que los puntos obtenidos con el método del Cedazo discriminan mejor la información no necesaria, aunque para su determinación tome más tiempo. Por otro lado creemos que la cantidad de puntos obtenidos por ORB puede ser mayor y probablemente tengamos que considerar otras opciones para la organización de los datos, lo cual aún no hemos explorado.

Las alternativas aquí presentadas nos proveen dos formas distintas para obtener la firma

⁹ rBRIEF es la versión modificada de BRIEF la cual toma en cuenta el ángulo [54] de los puntos detectados en FAST

espectral. Sin embargo antes es necesario realizar un pre-procesamiento de la información lo cual explicamos en la siguiente sección.

6.5 Pre-procesamiento de los datos

Una vez determinados los puntos de mayor información del conjunto de imágenes, ya sea por el método del Cedazo o por ORB, realizamos el acondicionamiento de los datos. El cual consiste en:

1. Calcular los histogramas para todas las imágenes utilizando los puntos obtenidos. Con esta información se crea una base de datos, en la cual las columnas o “bins” (variables independientes) corresponden a un intervalo del histograma¹⁰. En nuestro caso los “bins” van de 0 a 255 en incrementos de 1 en 1. Cada tupla en la base de datos corresponde a un histograma de una imagen (variable dependiente).
2. Hacer un análisis de correlaciones¹¹ para determinar si hay variables que se puedan eliminar de acuerdo a un umbral predeterminado. Nosotros escogimos un umbral del 95%.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1.00	0.82	0.97	0.88	0.86	0.96	0.95	0.94	0.91	0.88	0.87	0.83	0.82	0.86	0.91
2	0.82	1.00	0.91	0.73	0.52	0.77	0.92	0.77	0.72	0.81	0.90	0.90	0.86	0.90	0.73
3	0.97	0.91	1.00	0.84	0.76	0.92	0.98	0.92	0.88	0.88	0.91	0.88	0.86	0.91	0.88
4	0.88	0.73	0.84	1.00	0.89	0.90	0.81	0.91	0.91	0.97	0.79	0.84	0.84	0.86	0.92
5	0.86	0.52	0.76	0.89	1.00	0.92	0.73	0.90	0.92	0.85	0.69	0.67	0.70	0.72	0.91
6	0.96	0.77	0.92	0.90	0.92	1.00	0.92	0.98	0.97	0.93	0.87	0.85	0.86	0.89	0.97
7	0.95	0.92	0.98	0.81	0.73	0.92	1.00	0.90	0.85	0.88	0.96	0.93	0.91	0.95	0.87
8	0.94	0.77	0.92	0.91	0.90	0.98	0.90	1.00	0.99	0.92	0.82	0.83	0.83	0.87	0.98
9	0.91	0.72	0.88	0.91	0.92	0.97	0.85	0.99	1.00	0.92	0.78	0.80	0.81	0.84	0.98
10	0.88	0.81	0.88	0.97	0.85	0.93	0.88	0.92	0.92	1.00	0.88	0.93	0.94	0.95	0.94
11	0.87	0.90	0.91	0.79	0.69	0.87	0.96	0.82	0.78	0.88	1.00	0.97	0.96	0.97	0.82
12	0.83	0.90	0.88	0.84	0.67	0.85	0.93	0.83	0.80	0.93	0.97	1.00	0.99	1.00	0.85
13	0.82	0.86	0.86	0.84	0.70	0.86	0.91	0.83	0.81	0.94	0.96	0.99	1.00	0.99	0.86
14	0.86	0.90	0.91	0.86	0.72	0.89	0.95	0.87	0.84	0.95	0.97	1.00	0.99	1.00	0.89
15	0.91	0.73	0.88	0.92	0.91	0.97	0.87	0.98	0.98	0.94	0.82	0.85	0.86	0.89	1.00

Figura 6. Tabla de correlaciones para 15 variables. 9 variables están correlacionadas al 95%

3. Escalar los valores de todas la variables dentro del rango [0 – 1]. Por un lado hace para que todas las variables tengan la misma importancia relativa. Por otro al usar utilizar redes neuronales es necesario que la salida este dentro del rango de salida de la función de activación. Para este problema nosotros escogimos la función logística¹² que tiene cuyo rango de dominio es de [0 – 1]. Véase figura 7.

¹⁰ El número de columnas puede variar si el número de intervalos en el histograma es diferente.

¹¹ En el pre-procesamiento de los datos se utilizó la correlación de Pearson[64] definida como:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 (y_i - \bar{y})^2}}$$

donde: n

es el número de elementos que forman el conjunto de valores a analizar x_i y y_i son cada uno de los valores de las variables, \bar{x} y \bar{y} son las medias de las variables.

¹² La función logística esta definida como

$$y_i = \frac{1}{1 + \exp(-av_j)}$$

No escalados						Escalados					
V1	V2	V3	V4	V5	Y	V1	V2	V3	V4	V5	Y
5.0000	0.0000	0.0000	1.0000	1.0000	0.0000	0.5000	0.0000	0.0000	0.0625	0.1250	0.0000
1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2.0000	2.0000	11.0000	2.0000	8.0000	1.0000	0.2000	0.3333	1.0000	0.1250	1.0000	0.2500
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2500
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.2500
6.0000	0.0000	3.0000	1.0000	3.0000	2.0000	0.6000	0.0000	0.2727	0.0625	0.3750	0.5000
0.0000	0.0000	0.0000	0.0000	0.0000	2.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
0.0000	0.0000	0.0000	0.0000	0.0000	2.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.5000
10.0000	1.0000	1.0000	0.0000	3.0000	3.0000	1.0000	0.1667	0.0909	0.0000	0.3750	0.7500
0.0000	0.0000	1.0000	0.0000	0.0000	3.0000	0.0000	0.0000	0.0909	0.0000	0.0000	0.7500
0.0000	0.0000	0.0000	0.0000	0.0000	3.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.7500
9.0000	6.0000	6.0000	16.0000	6.0000	4.0000	0.9000	1.0000	0.5455	1.0000	0.7500	1.0000
0.0000	0.0000	0.0000	0.0000	0.0000	4.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000
0.0000	0.0000	0.0000	0.0000	0.0000	4.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Figura 7. Ejemplo del escalamiento de los datos para un problema de 5 variables independientes.

4. Estabilizar los datos para evitar que se presenten dependencias lineales que hagan que el sistema de ecuaciones que modela el fenómeno se vuelva inestable. La inestabilidad surge cuando hay una gran cantidad de valores idénticos. Esto típicamente ocurre cuando existen valores enteros que se repiten en los datos, como es en nuestro caso donde puede existir la misma cantidad de píxeles en distintas intensidades de color. Véase la figura 8.

No estabilizados						Estabilizados					
V1	V2	V3	V4	V5	Y	V1	V2	V3	V4	V5	Y
0.5000	0.0000	0.0000	0.0625	0.1250	0.0000	0.500000433	0.000000025	0.000000493	0.062500046	0.125000088	0.000000405
0.1000	0.0000	0.0000	0.0000	0.0000	0.0000	0.100000090	0.000000213	0.000000471	0.000000321	0.000000185	0.000000619
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.000000756	0.000000301	0.000000765	0.000000533	0.000000384	0.000000624
0.2000	0.3333	1.0000	0.1250	1.0000	0.2500	0.200000133	0.333333520	1.000000326	0.125000073	1.000000352	0.250000067
0.0000	0.0000	0.0000	0.0000	0.0000	0.2500	0.000000139	0.000000083	0.000000871	0.000000373	0.000000161	0.250000030
0.0000	0.0000	0.0000	0.0000	0.0000	0.2500	0.000000130	0.000000417	0.000000654	0.000000210	0.000000994	0.250000146
0.6000	0.0000	0.2727	0.0625	0.3750	0.5000	0.600000348	0.000000605	0.272727327	0.062500032	0.375000361	0.500000380
0.0000	0.0000	0.0000	0.0000	0.0000	0.5000	0.000000754	0.000000718	0.000000729	0.000000393	0.000000268	0.500000051
0.0000	0.0000	0.0000	0.0000	0.0000	0.5000	0.000000998	0.000000518	0.000000546	0.000000921	0.000000595	0.500000031
1.0000	0.1667	0.0909	0.0000	0.3750	0.7500	1.000000028	0.166666731	0.090909150	0.000000849	0.375000288	0.750000591
0.0000	0.0000	0.0909	0.0000	0.0000	0.7500	0.000000674	0.000000728	0.090909108	0.000000569	0.000000278	0.750000283
0.0000	0.0000	0.0000	0.0000	0.0000	0.7500	0.000000999	0.000000747	0.000000390	0.000000155	0.000000032	0.750000182
0.9000	1.0000	0.5455	1.0000	0.7500	1.0000	0.900000430	1.000000748	0.545454857	1.000000698	0.750000204	1.000000359
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.000000777	0.000000017	0.000000227	0.000000534	0.000000004	1.000000029
0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	0.000000396	0.000000917	0.000000458	0.000000129	0.000000715	1.000000308

Figura 8. Ejemplo de la estabilización de datos.

Los datos originales se perturban aleatoriamente de acuerdo a la siguiente regla:

$$v_i^* = \begin{cases} v_i \times (1 + \rho \cdot \delta_H) & \text{if } v_i \neq 0 \\ \rho \cdot \delta_H & \text{if } v_i = 0 \end{cases} \quad (3)$$

donde:

donde a es un parámetro que controla la pendiente de la función y v_i es el campo local inducido (i.e. la suma de todas las entradas a la neurona) [20]

v_i es el valor original de las variables, ρ_u es un valor aleatorio entre $0 \leq \rho_u < 1$ de una distribución uniforme, δ_H es un valor muy pequeño de $O(10^{-6})$ y v_i^* es el valor alterado. Al resolver un problema de regresión utilizando estos nuevos datos, no estamos obteniendo la solución al problema original, sólo estamos aproximándonos a la solución requerida. En [37] se demostró que la diferencia relativa entre la función de aproximación de los valores originales (f) y el que corresponde a los datos estabilizados (f') es $O(\delta_H)$. Por lo tanto $|f - f'|/|f| < \delta_H$. A continuación trascibimos su derivación matemática:

Sea N el conjunto de datos originales, N^* el conjunto de datos resultante después de realizar la perturbación. M_τ y M_τ^* ¹³ son conjunto interno objetivo para N y N^* respectivamente. v_τ y v_τ^* denotan cualquier vector en los conjunto objetivo M_τ y M_τ^* para N y N^* respectivamente. Establecemos una cota en la diferencia entre la aproximación minimax a N^* en v_τ^* y en v_τ . Sea

$$F_\tau^* = \sum_{i_1=0}^{g_1} \cdots \sum_{i_p=0}^{g_p} C_{i_1 \cdots i_p} \left(\left(v^* \right)_1^{i_1} \right)_\tau \cdots \left(\left(v^* \right)_p^{i_p} \right)_\tau \quad (4)$$

el valor del polinomio minimax obtenido de N^* y evaluado en cualquier punto del conjunto M_τ^* ; sea

$$F_\tau = \sum_{i_1=0}^{g_1} \cdots \sum_{i_p=0}^{g_p} C_{i_1 \cdots i_p} \left(v_1^{i_1} \right)_\tau \cdots \left(v_p^{i_p} \right)_\tau \quad (5)$$

el valor del polinomio minimax obtenido de N y evaluado en cualquier punto del M_τ ; y $TDG(x) \equiv$ términos de grado x en F_τ ; $TDO(x) \equiv$ términos de orden x en F_τ . Entonces,

$$\begin{aligned} F_\tau^* = & (1 + \delta_H)^0 x TDG(0) + (1 + \delta_H)^1 x TDG(1) + \\ & \cdots + (1 + \delta_H)^{g_1 + \cdots + g_p} x TDG(g_1 + \cdots + g_p) \end{aligned} \quad (6)$$

en donde, tomamos δ_H como el valor máximo de δ_i . Al expandir los binomios tenemos:

$$\begin{aligned} F_\tau^* = & TDG(0) + \cdots + TDG(g_1 + \cdots + g_p) + \\ & \delta_H \left[1 x TDG(1) + \cdots + (g_1 + \cdots + g_p) x TDG(g_1 + \cdots + g_p) \right] \\ & + TDO(\delta_H^2) + \cdots + TDO(\delta_H^{g_1 + \cdots + g_p}) \end{aligned} \quad (7)$$

Ya que $\delta_H \ll 1$ podemos escribir,

¹³ El subíndice τ denota el conjunto objetivo.

$$\begin{aligned}
F_{\tau}^* &\approx TDG(0) + \dots + TDG(g_1 + \dots + g_p) + \\
&\quad \delta_H [1xTDG(1) + \dots + (g_1 + \dots + g_p)xTDG(g_1 + \dots + g_p)] \\
F_{\tau}^* - F_{\tau} &= \delta_H [1xTDG(1) + \dots + (g_1 + \dots + g_p)xTDG(g_1 + \dots + g_p)]
\end{aligned} \tag{8}$$

y, sucesivamente,

$$\begin{aligned}
\delta_H \sum_{i=1}^{g_1+\dots+g_p} ixTDG(i) &= \delta_H \left\{ \left[F_{\tau} - TDG(0) \right] + \left[F_{\tau} - TDG(0) - TDG(1) \right] + \dots \right. \\
&\quad \left. + \left[F_{\tau} - TDG(0) - TDG(1) - \dots - TDG(g_1 + \dots + g_p - 1) \right] \right\} \\
&= \delta_H \sum_{i=1}^{g_1+\dots+g_p} \left\{ F_{\tau} - \sum_{j=0}^{i-1} TDG(j) \right\}
\end{aligned} \tag{9}$$

tomando valores absolutos

$$\left| F_{\tau}^* - F_{\tau} \right| \approx \left| \delta_H \sum_{i=1}^{g_1+\dots+g_p} \left[F_{\tau} - \sum_{j=0}^{i-1} TDG(j) \right] \right| \tag{10}$$

puesto que $\delta_H > 0$ y $\left| \sum_i a_i \right| \leq \sum_i |a_i|$ entonces

$$\left| F_{\tau}^* - F_{\tau} \right| \leq \delta_H \sum_{i=1}^{g_1+\dots+g_p} \left| F_{\tau} - \sum_{j=0}^{i-1} TDG(j) \right| < \delta_H x(g_1 + \dots + g_p) |F_{\tau}| \tag{11}$$

de donde concluimos

$$\left| \frac{F_{\tau}^* - F_{\tau}}{F_{\tau}} \right| < \left[\sum_i g_i \right] \delta_H \tag{12}$$

Por otra parte podemos escribir

$$\begin{aligned}
(i) \quad f_{\tau} - F_{\tau} &= \varepsilon_{\theta} \\
(ii) \quad f_{\tau} + f_{\tau} \delta_H - F_{\tau}^* &= \varepsilon_{\theta}^*
\end{aligned} \tag{13}$$

Restando (ii) de (i)

$$F_{\tau}^* - F_{\tau} - f_{\tau} \delta_H = \varepsilon_{\theta} - \varepsilon_{\theta}^* \tag{14}$$

Sabemos de la ecuación (10) que

$$F_{\tau}^* - F_{\tau} - f_{\tau}\delta_H = \delta_H \left[\sum_{i=1}^{g_1+\dots+g_p} \left\{ F_{\tau} - \sum_{j=0}^{i-1} TDG(j) \right\} - f_{\tau} \right] \quad (15)$$

Por lo tanto

$$\begin{aligned} |\varepsilon_{\theta} - \varepsilon_{\theta}^*| &\leq \delta_H \left\{ \sum_{i=1}^{g_1+\dots+g_p} \left| F_{\tau} - \sum_{j=0}^{i-1} TDG(j) \right| + |f_{\tau}| \right\} \\ |\varepsilon_{\theta} - \varepsilon_{\theta}^*| &< \delta_H \left\{ (g_1 + \dots + g_p) |F_{\tau}| + |f_{\tau}| \right\} \end{aligned} \quad (16)$$

y como $|f_{\tau}| < |F_{\tau}|/2$; $\varepsilon_{\theta} > 0$ y $\varepsilon_{\theta}^* > 0$ entonces

$$|\varepsilon_{\theta} - \varepsilon_{\theta}^*| < \left(2 + \left[\sum_i g_i \right] \right) \delta_H |F_{\tau}| \quad (17)$$

Con las ecuaciones (12) y (17) se puede concluir que al estabilizar los datos utilizando este proceso de perturbación lleva a una diferencia relativa entre la función evaluada en N^* y la función evaluada en N del orden de δ_H . Por otra parte, la diferencia de los errores minimax (ε_{θ}) en N^* y N es, también, del orden de δ_H . Puesto que el valor de δ_H es, por definición, muy chico, la función de aproximación que se obtiene del conjunto perturbado aleatoriamente es muy cercano a la función de aproximación original obtenido de N .

5. Mejorar y complementar los datos utilizando *splines* naturales [55] en caso que no se cuente con suficientes datos o estén incompletos. Esto es particularmente importante porque los fundamentos teóricos¹⁴ en los que están basadas las redes de perceptrones multicapa, que utilizamos para realizar la aproximación multivariada y que explicaremos más adelante, implica que tiene que haber continuidad en los datos. Al utilizar *splines* naturales tenemos un polinomio de colocación para los datos conocidos con la mínima curvatura. En la figura 9 se muestran 60 datos obtenidos por medio de un *spline* natural para 15 datos conocidos.

¹⁴ Estos fundamentos se basan en la prueba del Teorema de Aproximación Universal, Universal Approximation Theorem (UAT), de Cybenko [15].

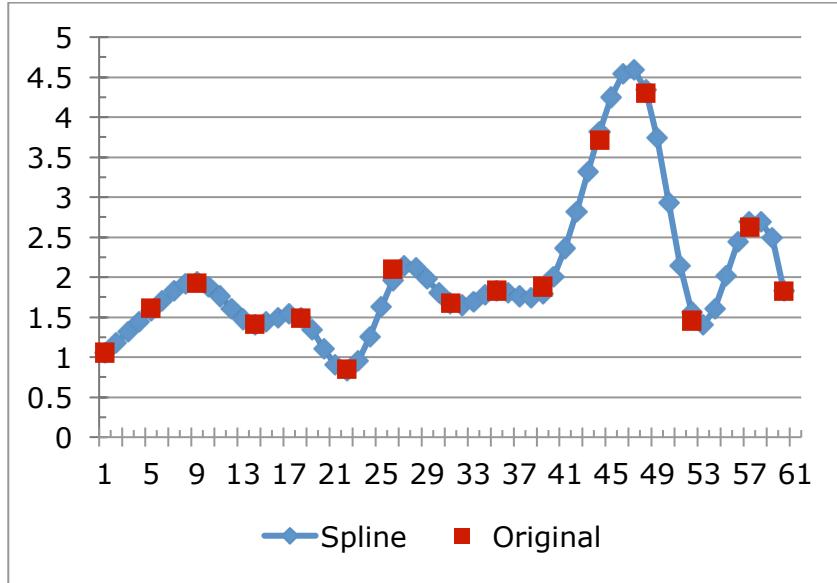


Figura 9. *Spline* natural que coloca los datos originales.

6. Determinar la cantidad de información contenida en los datos. Esto es importante para 1) determinar la muestra mínima de la información necesaria y en el caso de las Redes de Perceptrones 2) es un dato requerido para determinar su arquitectura. Estrictamente hablando la cantidad de información corresponde a la complejidad de Kolmogorov, no obstante se sabe que esta es incomputable [39] dado que consiste en encontrar la expresión mas compacta de un programa para una máquina universal de Turing que reproduzca los datos observados. Sin embargo, es posible aproximarse a dicha complejidad usando un buen algoritmo de compresión [11]. En nuestro caso utilizamos el algoritmo de PPM (Prediction by Partial Matching) [2, 58]. La tasa de compresión (I) obtenida determina la cantidad de información real que contiene la base datos de acuerdo a la siguiente fórmula.

$$N' = \frac{N}{I} \quad (18)$$

En donde N es el número de bytes total de información y N' es el número de bytes que corresponde a la cantidad real de información.

Una vez ejecutados estos pasos, los datos están listos para ser utilizados en cualquiera de los métodos descritos a continuación para buscar el aproximador. Cuyo objetivo, como ya mencionamos, es en encontrar lograr la identificar de las imágenes en la fase de navegación.

6.6 Aproximación Multivariada

Como mencionamos al final de la sección 6.3, originalmente para identificar las imágenes con los puntos obtenidos por el método del Cedazo se utilizaron diferentes medidas de similitud. Estas medidas de similitud permiten comparar directamente el valor de los píxeles pertenecientes al cedazo de la imagen a identificar contra las imágenes de referencia. Sin embargo este problema puede verse como un problema de aprendizaje supervisado. En

donde el objetivo es encontrar una expresión de una variable dependiente (y) en función de un conjunto de n variables independientes (\mathbf{v}), tal que:

$$y = f(v_1, v_2, \dots, v_n) \quad (19)$$

La función $f(\mathbf{v})$ mantendrá los “patrones genéricos” presentes en la colección de datos. Si logramos encontrar dicha función f , podremos identificar una imagen dados ciertos valores de \mathbf{v} . El problema puede parecer trivial cuando el número de variables es igual a uno¹⁵. Sin embargo no lo es cuando el número de variables es mayor. Como ya mencionamos, para determinar la mejor forma de encontrar la función f se analizaron [41] dos enfoques: 1) Métodos de aproximación multivariada basados en consideraciones algebraicas y 2) Redes de Perceptrones. Ambos fueron sometidos a una comparación para determinar: a) cuál de los algoritmos basados en el enfoque (1) se desempeña mejor y b) determinar si son equiparables a las Redes de Perceptrones.

En el enfoque de aproximación multivariada deseamos encontrar un aproximador de la forma:

$$y = c_1 X_1 + c_2 X_2 + \dots + c_m X_m \quad (20)$$

donde X_i es una combinación arbitraria de las n variables independientes. Es decir, $X_i = f_i(\mathbf{v})$. Dependiendo de la forma en la que estas combinaciones se definen se puede obtener diferentes aproximadores. Para resolver este problema se requiere encontrar un conjunto de coeficientes c (de tamaño m) sobre el conjunto de las n variables independientes expresadas como una combinación lineal de monomios¹⁶ o términos de las n variables de grado $d_1 \dots d_n$, tal que la diferencia, utilizando la norma L_2 ó L_∞ , entre la función de aproximación y los datos observados sea minimizada. De antemano sabemos que la norma L_∞ es susceptible a datos atípicos (outliers) y que la norma L_2 es inmune a estos. Sin embargo, Cheney mostró que cuando no hay datos atípicos en el conjunto de datos a analizar, ambas normas son equivalentes [8]. Además, como ya mencionamos, cuando se utilizan este tipo de métricas se presentan 2 problemas 1) el problema de alta dimensionalidad y 2) la memoria requerida para obtener el modelo. Nuestra intención al realizar este análisis es i) determinar cuál de las dos normas resulta ser la mejor para el problema que queremos resolver, ii) si los algoritmos se pueden aplicar para obtener el modelo y iii) cuál de ellos es la mejor opción. Para poder saber esto examinamos tres algoritmos: a) el Algoritmo de Levenberg-Marquardt (LM) [38, 49], b) el Algoritmo de Powell (PM) [47] y c) el Algoritmo de Rápido Ascenso (Fast Ascent Algorithm o FA) [8]. Los algoritmos (a) y (b) minimizan la norma L_2 :

$$F(\mathbf{v}) = \frac{1}{2} \sum_{i=1}^m (y_i - f_i)^2 \quad (21)$$

¹⁵ CurveExpert Basic[25] es un software de regresión bivariada que incorpora métodos de regresión lineal y no lineal. Por un lado los métodos de regresión lineal encuentran un modelo polinomial resolviendo un sistema de ecuaciones lineales. Por otro lado los métodos de regresión no-lineal utilizan el algoritmo de Levenberg-Marquardt.

¹⁶ Cada X_i corresponde a un monomio. La combinación lineal de los monomios es $y = \sum c_i X_i$; $i = 1, 2, \dots, m$.

Mientras que (c) minimiza la norma L_∞ :

$$F(\mathbf{v}) = \max |y_i - f_i| \text{ donde } i = 1, 2, \dots, m. \quad (22)$$

Nótese que en la ecuación (21) debemos tener en memoria el conjunto completo de los N datos. A continuación describiremos brevemente los algoritmos. El detalle de cada uno junto con su pseudocódigo se encuentra en el anexo 2.

6.6.1 Algoritmo de Levenberg-Marquardt

Es un proceso iterativo donde se combinan el a) método de Gauss-Newton y b) el gradiente descendiente. El algoritmo explora el espacio de soluciones a partir de un vector inicial, con el cual se resuelve un sistema de ecuaciones que involucra el conjunto completo de los N datos. El vector \mathbf{x} se aproxima a la solución por medio de un parámetro de amortiguamiento μ . El parámetro μ determina cómo se comporta el algoritmo. Si μ se approxima a 0 , el algoritmo se comporta como (a) si μ se incrementa se approxima a (b).

6.6.2 Algoritmo de Powell

El algoritmo es muy similar al de LM, sin embargo en lugar de un parámetro de amortiguamiento, utiliza una región de confianza para controlar la aproximación a la solución. De igual forma que LM, también necesita del conjunto completo de N datos para resolver el sistema de ecuaciones.

6.6.3 Algoritmo de Rápido Ascenso

Es un proceso iterativo de dos fases. En la primer fase se selecciona aleatoriamente una muestra de tamaño M (llamado conjunto interno) y se obtiene el mejor conjunto de coeficientes del aproximador que satisfacen la norma minimax. En la segunda etapa se prueba el aproximador con el conjunto N-M datos restantes (conjunto externo). Si se mantiene la condición minimax con todos los N-M datos restantes el algoritmo termina. De lo contrario se selecciona apropiadamente un elemento de ambos conjuntos para intercambiarlos y el proceso se repite. Una característica de este método es que sin importar el tamaño del conjunto de datos únicamente se tienen que resolver sistemas de ecuaciones donde además de resolver para el número de coeficientes deseados (m) en el aproximador, también resolvemos para el error. Por lo que el tamaño del sistema de ecuaciones a resolver es $M' \times M'$ donde $M' = m + 1$.

El número de términos del aproximador obtenido por los tres algoritmos está directamente relacionado con el número de variables independientes y el grado que se le asigne a cada una de ellas. Este tipo de problemas son problemas de alta dimensionalidad ya que el número de coeficientes en la solución crece exponencialmente. Por ejemplo para un problema que consista de 8 variables independientes y asignando a cada una un grado máximo de 2 el número de términos en el polinomio aproximador será de 6,561.

6.7 Polinomios genéticos multivariados

El problema de alta-dimensionalidad se puede resolver incorporando un Algoritmo Genético a los algoritmos antes descritos para seleccionar un subconjunto del total de coeficientes que forman el aproximador. A esta solución se le conoce como Polinomios Genéticos Multivariados [30]. Los parámetros para encontrar el Polinomio Genético son: a) el número de

términos deseados en el modelo y b) el grado máximo de cada una de las variables independientes.

Para este estudio [41] seleccionamos un conjunto problemas tanto de clasificación como de regresión del repositorio de aprendizaje de máquina de la universidad de California en Irving¹⁷. En la tabla 2 se muestra el tipo de problema a resolver, el número de variables y datos que conforman la base de datos. A cada uno de estos problemas se les aplicó el preprocesamiento de datos descrito en al sección 6.5. El número de atributos para cada problema, mostrados en la misma tabla, corresponde al número de variables independientes reportadas originalmente por los autores de los datos. Sin embargo algunas de estas variables son categóricas. Por lo tanto es necesario agregar el siguiente paso al preprocesamiento de los datos:

- La variables categóricas de deben codificar determinando si: a) Se reemplaza cada variable categórica por un conjunto de variables binarias “dummy” donde cada nueva variable corresponde a una instancia de la categoría, o b) se establece un código válido para cada instancia de los valores categóricos.

Si elegimos (a), el número de variables aumenta considerablemente. Por ejemplo, el número de variables del problema “servo” en la tabla 2, originalmente consta de 4 variables independientes. Sin embargo todas son categóricas. Al reemplazarlas por variables binarias “dummy”, el número de variables independientes aumenta a 17 como se puede ver en la tabla 3 de la sección 6.7. Si elegimos (b), los códigos que reemplacen a cada instancia de las variables categóricas deben preservar las relaciones originales de los datos sin introducir nuevos patrones en estas, un ejemplo de este tipo de enfoque se puede ver en [34].

Para este conjunto de problemas decidimos reemplazar cada variable categórica por un conjunto de variables binarias “dummy”.

Tabla 2. Descripción de los problema seleccionados del repositorio de aprendizaje de máquina de UCI

Nombre	Tipo	# Registros	# Atributos
Climate Model	Clasificación	540	18
Breast Cancer	Clasificación	699	11
Wine	Clasificación	178	13
Red Wine quality	Clasificación	1599	12
White Wine quality	Clasificación	4898	12
Housing	Regresión	506	14
CPU Performance	Regresión	209	10
Servo	Regresión	167	4

Los parámetros asignados para resolver los 8 problemas están resumidos en la tabla 3. En todos los problemas se utilizó el algoritmo EGA. La probabilidad de cruzamiento $P_c = 1.0$ y de mutación $P_m = 0.01$ permanecieron fijas en todo momento. Las columnas de la tabla 3 muestran el tipo de problema, el número de variables independientes después del preprocesamiento, el número de términos deseados en el polinomio, el grado máximo que puede tomar cada una de las variables independientes, el tamaño de la población, el número máximo de generaciones y el número de total de términos que conforman el polinomio completo.

¹⁷ UC Irvine Machine Learning Repository [40][14]

Tabla 3. Parámetros utilizados en los Experimentos para encontrar el Polinomio Genético Multivariado.

Nombre	T ^a	IV ^b	Términos deseados	D ^c	Ind ^d	Gen	Total Términos
Climate	C	21	60	1	25	200	2,097,152
Breast Cancer	C	7	12	1	25	300	128
Red Wine quality	C	11	15	1	20	100	2,048
White Wine quality	C	11	15	1	20	100	2,048
Wine	C	13	12	1	30	100	8,192
Housing	R	12	9	1	40	200	4,096
CPU Performance	R	8	12	2	75	300	6,561
Servo	R	17	25	1	50	500	131,072

^a Tipo de problema. C para Clasificación y R para Regresión

^b Variables Independientes

^c Grado de cada variable independiente en el modelo

^d Número de individuos en la población

Los individuos que conforman la población están codificados en una cadena que representa los términos del polinomio propuesto y los grados de cada una de las variables. Por lo tanto el tamaño del genoma está determinado por el número de monomios deseados y el número de variables independientes. Cada elemento en el genoma corresponde al grado que va tomar la variable dentro del monomio. Al codificar los individuos de esta forma no es necesario calcular todas las posibles combinaciones de monomios. En su lugar EGA da la mejor propuesta del polinomio del espacio de posibles soluciones. En la figura 10 mostramos un ejemplo del genoma para un polinomio aproximador compuesto de 5 términos y 3 variables independientes de grado máximo 2.

Términos				
Variables			Genoma	
0	1	2	3	4
0	1	2	0	1
1	2	2	2	0
2	2	2	0	2
0	2	1	2	2
2	1	2	0	1
0	1	0	2	1

Figura 10. Ejemplo del genoma de un individuo para un polinomio genético. El genoma consta 5 términos y 3 variables independientes. Cada variable tiene grado máximo de 2.

Los operadores del EGA son: selección determinística, cruzamiento anular o en dos puntos, mutación uniforme y elitismo completo. Para evitar la convergencia prematura se utiliza una estrategia en dos fases: a) se seleccionan $2n$ individuos que corresponden a las últimas 2 generaciones y se ordenan del mejor al peor. Se seleccionan los mejores n individuos para que sobrevivan. b) Para el cruzamiento se realiza entre el individuo i y el individuo $n-i+1$ para generar nuevos individuos. En cada generación los individuos sobrevivientes se convierten en los mejores individuos de todo el proceso.

Con los coeficientes encontrados por los algoritmos utilizando cada una de las normas se cálculo del error de aproximación. Comparando los errores de aproximación, tablas 4 y 5, FA resulta ser el mejor cuando se minimiza la norma L_∞ , mientras que al minimizar la norma L_2 , PM es el que obtiene mejores aproximaciones.

Tabla 4. Error de aproximación L_2

Nombre	FA	LM	PM
Climate	0.316189	0.211406	0.153626
Breast Cancer	0.234506	0.198531	0.197543
Red Wine quality	0.134421	0.1556166	0.124052
White Wine quality	0.142363	0.1401591	0.119953
Wine	0.112512	0.119995	0.091971
Housing	0.100802	0.0833473	0.078231
CPU Performance	0.023816	0.0193315	0.019116
Servo	0.139994	0.0964289	0.067942

Tabla 5. Error de aproximación L_∞

Nombre	FA	LM	PM
Climate	0.541581	0.973984	1.011879
Breast Cancer	0.505585	1.009086	1.04041
Red Wine quality	0.33644	0.647101	0.489728
White Wine quality	0.446125	0.651144	0.470141
Wine	0.201608	0.429697	0.644723
Housing	0.288528	0.451093	0.454289
CPU Performance	0.045491	0.111987	0.087435
Servo	0.179541	0.325534	0.343606

En conclusión podemos ver que efectivamente tanto LM como PM que están basados en la norma L_2 son menos susceptibles a datos atípicos. Sin embargo, la cantidad de memoria que necesitan ambos algoritmos para resolver los sistemas de ecuaciones es mayor debido a que requieren usar todos los datos del problema de manera simultánea. Esto es un problema cuando se cuenta recursos de memoria limitados. En contraste FA, basado en L_∞ , solo requiere de un subconjunto de los datos, igual al número de coeficientes del aproximador m mas uno ($M = m + 1$), para resolver el sistema de ecuaciones. Esto hace que la cantidad de memoria requerida en FA sea mucho menor. Tomando en cuenta que ambas normas son equivalentes cuando no existen datos atípicos, como ya mencionamos al inicio de la sección 6.6, y considerando que típicamente en el problema que queremos resolver estos no se existen, creemos que FA es el mejor algoritmo para nuestros propósitos. Sin embargo, para tener un análisis completo comparamos estos resultados contra las Redes de Perceptrones lo cual mostramos en la siguiente sección.

6.8 Redes de Perceptrones Multicapa (MLPs)

El propósito de analizar este enfoque es determinar si los algoritmos de aproximación multivariada basado en consideraciones algebraicas son equiparables comparados con las MLPs, tomando en cuenta que se ha probado teóricamente que las MLPs son aproximadores universales [15, 22, 23]. A diferencia de los algoritmos descritos en las secciones 6.6 y 6.7 en donde se obtiene un polinomio que aproxima a la función original, las MLPs son una “caja-negra”, es decir no se obtiene un modelo algebraico. Para las MLPs se define una

arquitectura de la red y se calcula una colección de valores para las conexiones de la red, llamados “pesos”.

Resolvimos los mismos problemas descritos en la tabla 3 para hacer la comparación. Las configuraciones de la red se determinaron cómo se indica en la sección 6.8.1. Estas, así como el error de aproximación obtenido para cada problema se muestran en la tabla 6.

Tabla 6. Configuración de las MLP y el error de entrenamiento.

Nombre	#H ^a	Cons ^b	Épocas	L_{∞}	L_2
Climate	6	132	6,000	0.284628	0.017601
Breast Cancer	10	80	4,000	1.004898	0.088992
Red Wine quality	14	168	20,000	0.455825	0.087623
White Wine quality	34	408	2,000	0.477333	0.094080
Wine	5	70	4,000	0.123468	0.030756
Housing	7	91	4,000	0.454456	0.045501
CPU Performance	5	45	5,000	0.147365	0.020764
Servo	5	90	4,000	0.176295	0.017448

^a Número de nodos in la capa escondida.

^b Número de conexiones

Determinamos el desempeño de los algoritmos comparando el error obtenido tomando en cuenta, por un lado el número de términos en los polinomios de aproximación y por otro lado el número de conexiones en las MLPs. El resultado de esta comparación muestra que los aproximadores obtenidos por medio de los algoritmos basados en consideraciones algebraicas si son equivalentes a las MLPs. En las figuras 11 y 12 se muestra que las MLPs obtienen un mejor error de aproximación. Tomando en cuenta la escala del error en ambos casos, los aproximadores obtenidos son relativamente buenos. En especial, el FA se desempeña mejor que los otros 2 cuando minimiza la norma L_{∞} y su desempeño es similar cuando minimiza la norma L_2 . En conclusión la aproximación multivariada puede obtenerse por lo tanto de las MLP o por los GMP.

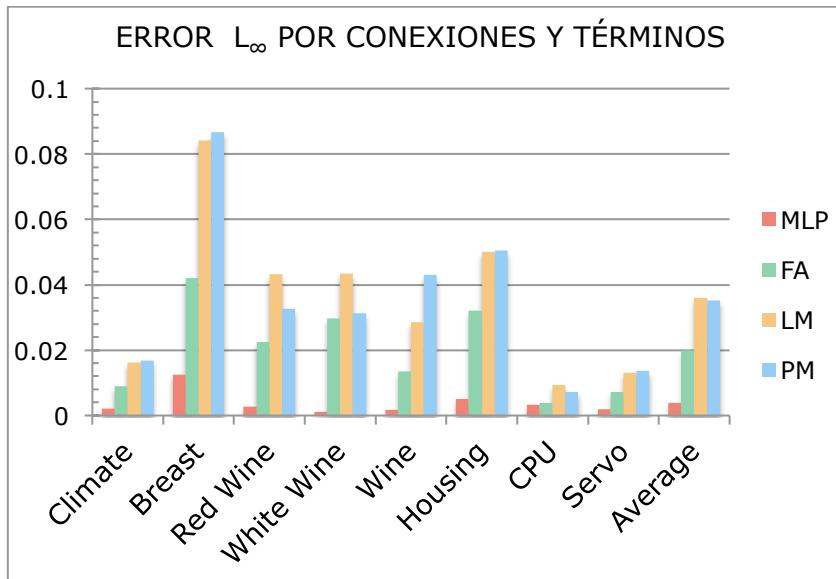


Figura 11. Desempeño de los algoritmos usando la norma L_{∞}

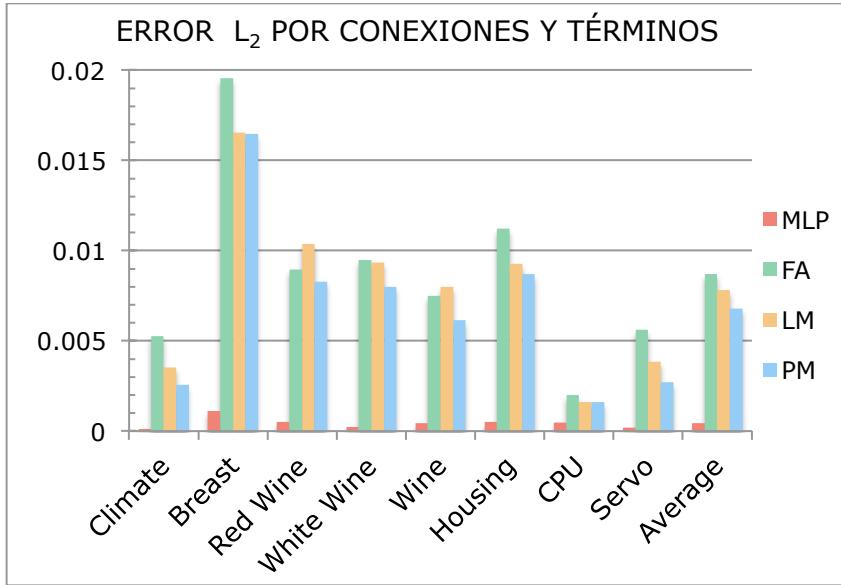


Figura 12. Desempeño de los algoritmos usando la norma L_2

6.8.1 Determinación de la arquitectura de las MLPs

En general la arquitectura de la red se desconoce y regularmente se determina de forma heurística. Sin embargo, existe una metodología para determinar la mejor arquitectura para las MLPs [28]. Los parámetros que se consideran para establecer la arquitectura de una MLP son: a) el número de neuronas de entrada m_0 , b) el número de capas escondidas, c) el número de neuronas en la capa escondida m_l , d) la función de activación $\varphi(\cdot)$, e) los pesos de las conexiones entre las neuronas w_{ji} y f) las neuronas en la capa de salida m_i . Gracias al teorema de aproximación universal (UAT) [20] se sabe que sólo es necesaria una capa escondida. Las consideraciones a las que se refiere el teorema son: 1) La función $\varphi(\cdot)$ debe ser continua, no-constante, acotada y monótonicamente creciente y 2) los datos deben estar en el intervalo de $[0,1]$. El número de neuronas de entrada m_0 lo conocemos a partir del número de variables de nuestros datos. Se seleccionó como función de activación la función logística. Sólo existe una neurona en la capa de salida que corresponde a la variable dependiente. Los pesos de las conexiones se determinan al entrenar a la red. Lo que queda por determinar es el número de neuronas de la capa escondida. Para calcular m_l es necesario calcular la información en los datos (N). El cálculo de N se realizó como se discutió en el paso 7 en la sección 6.5 de pre-procesamiento. Una vez teniendo ésta se puede aplicar la fórmula 22 derivada en [28] para obtener la cota inferior de m_l . Para poder aplicar (23) es necesario escalar los valores de m_0 , N , m_l en el intervalo de $[0,1]$. Los valores escalados se denotan con m_0^* , N^* , m_l^* . El rango utilizado para escalarlos se muestra en la tabla 7

Tabla 7. Rangos utilizados para escalar los valores de m_0 , N , m_l .

	m_0	N	m_l
max	82	25,000	549
min	2	100	0

$$\begin{aligned}
m_I^* = & c_{01}N^* + c_{12}(m_0^*)(N^*)^2 + c_{13}(m_0^*)(N^*)^3 + c_{15}(m_0^*)(N^*)^5 + c_{22}(m_0^*)^2(N^*)^2 + \\
& c_{24}(m_0^*)^2(N^*)^4 + c_{32}(m_0^*)^3(N^*)^2 + c_{35}(m_0^*)^3(N^*)^5 + c_{42}(m_0^*)^4(N^*)^2 + \\
& c_{45}(m_0^*)^4(N^*)^5 + c_{52}(m_0^*)^5(N^*)^2 + c_{63}(m_0^*)^6(N^*)^3
\end{aligned} \tag{23}$$

donde los subíndices de los coeficientes c_{ij} indican las potencias de las variables m_0^* y N^* respectivamente. Los valores de los coeficientes están dados en la tabla 8.

Tabla 8. Valores de los coeficientes para encontrar m_I .

Coeficiente	Valor	Coeficiente	Valor
c01	0.9307	c32	-209.393
c12	-33.6966	c35	52.818
c13	24.5008	c42	205.6786
c15	-3.4958	c45	-32.6766
c22	107.497	c52	-80.8771
c24	-41.0848	c63	9.7786

Una vez determinada la arquitectura, la red se entrena utilizando el algoritmo de retro-propagación. Al inicio del entrenamiento los pesos de las conexiones toman valores aleatorios y se van refinando de acuerdo al error obtenido de comparar el valor deseado con el obtenido a la salida en cada iteración.

7 Resultados

Con la metodología descrita, realizamos una prueba en un conjunto de 5 imágenes a color, distintas entre sí, de 144 píxeles de alto por 192 píxeles de ancho para un total de 27,648 píxeles por imagen. Las imágenes se extrajeron de un vídeo en donde el robot se mueve por un ambiente. El vídeo tiene una duración de 5:20 minutos y se grabó a 30 cuadros por segundo. Tomamos una imagen del vídeo cada 13 segundos para un total de 738 imágenes. La 5 imágenes con las que hicimos nuestra prueba se seleccionaron al azar del conjunto total de imágenes. Véase la figura 13.

Para la etapa de pre-entrenamiento se empleo la versión del método del Cedazo utilizando la norma L_∞ . El proceso se ejecuto usando el total de píxeles en la imagen (27,648). El cedazo resultante consiste de 6,369 píxeles (objetivo 1). Utilizando este método logramos un error de 0.000307 (objetivo 2). La tasa de compresión es de aproximadamente 4:1. Con los puntos determinados por el método de Cedazo se procedió obtener los valores de los píxeles para cada una de las imágenes. En total se van analizan 19,107 valores por cada canal de color (RGB) por imagen, en lugar de 82,944 valores por imagen. Con esta información calculamos el histograma de las imágenes para cada canal de color y se creó una base de datos experimental como se describió en la sección de pre-procesamiento. La base de datos consiste de 256 variables independientes, los “bins” del histograma, una variable dependiente, la imagen, y 15 tuplas, tres por cada imagen.

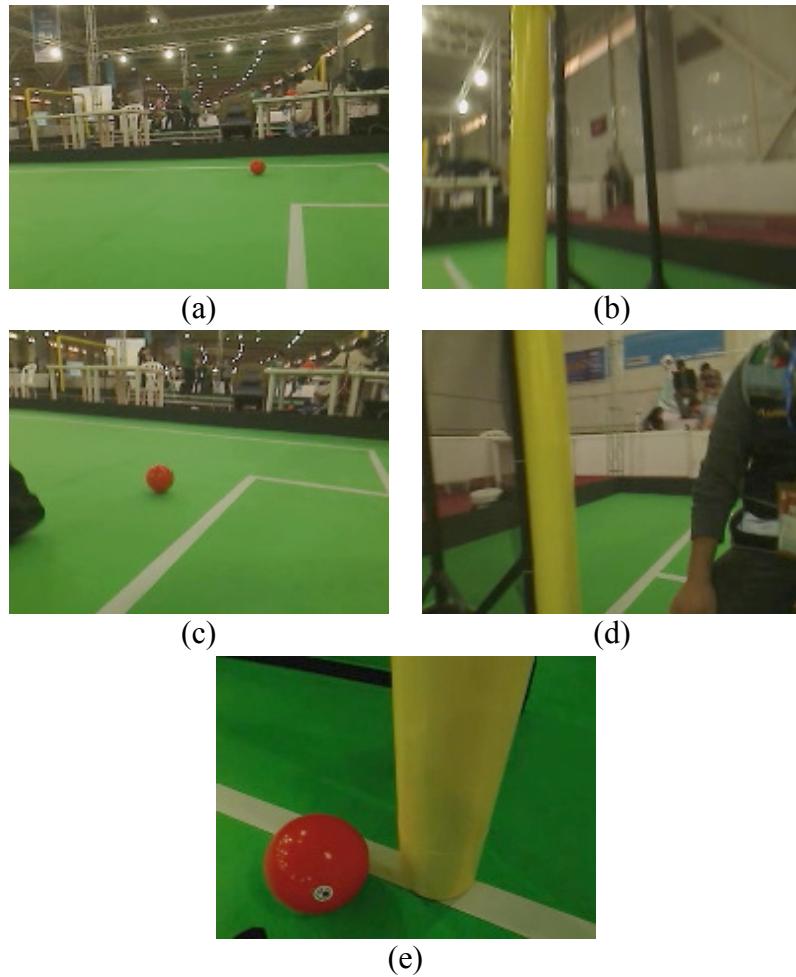


Figura 13. Imágenes utilizadas en la prueba de la metodología

Después del realizar el pre-procesamiento descrito en la sección 6.5 y de efectuar un análisis de correlaciones con un umbral del 95% el número de variables independientes se redujo de 256 a 156. Esta base de datos fue la que empleamos en la etapa de navegación usando los enfoques aquí expuestos. En los métodos aproximación multivariada basados en consideraciones algebraicas los polinomios resultantes no satisficieron nuestro mínimo determinado. La razón es que el número de variables es muy grande y como consecuencia el número de términos mínimos para resolver problema con el grado de precisión que requerimos es demasiado grande, i.e. el problema que vamos a resolver consta de 256 variables. Si consideramos que cada variable es de grado 1, el número de términos que podría contener el polinomio aproximador es del $O(2^{256})$, incluso después de reducir el número de variables independientes el orden es de $O(2^{156})$. Aún utilizando Polinomios Genéticos Multivariados el número de términos que se necesitan para lograr una aproximación aceptable es demasiado grande.

Debido a esta razón utilizamos las MLP. Para este enfoque probamos 2 arquitecturas de MLPs. La primera usando el total de variables independientes (256) antes de hacer el análisis de correlaciones. La segunda arquitectura usando el número de variables (156) después del análisis de correlaciones. Nótese que el número de variables (m_0) en ambos casos es mayor al límite superior determinado en la tabla 7. Para poder utilizar la ecuación 22 tuvi-

mos que redefinir el límite superior para la variable m_0 . Los nuevos rangos utilizados se muestran en la tabla 9.

Tabla 9. Nuevo rangos utilizados para escalar los valores de m_0, N, m_I .

	M_0	N	m_I
max	256	25,000	549
min	2	100	0

Ambas arquitecturas requirieron 3 neuronas en la capa escondida. Ambas usaron la función logística en la capa escondida. Con ambas arquitectura logramos obtener una clasificación sin error, como se muestra en las figuras 14 y 15.

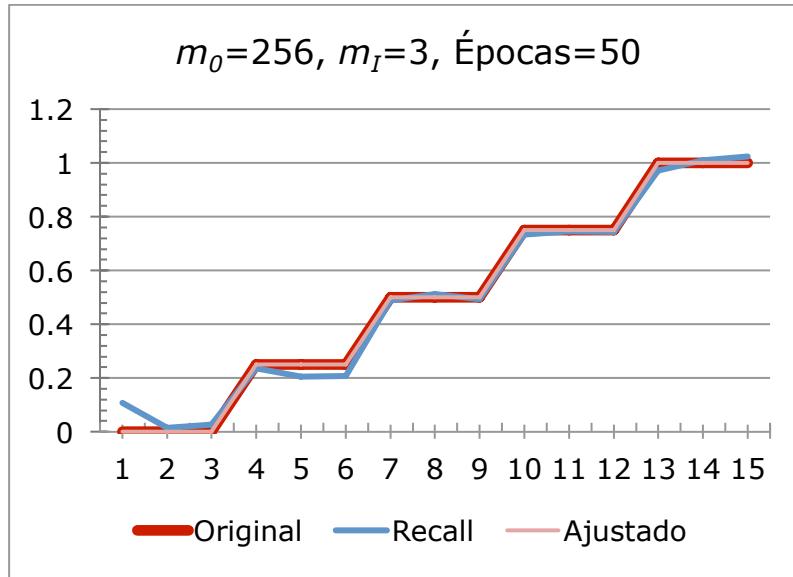


Figura 14. Salida de la clasificación de la imágenes utilizando 256 variables en MLP

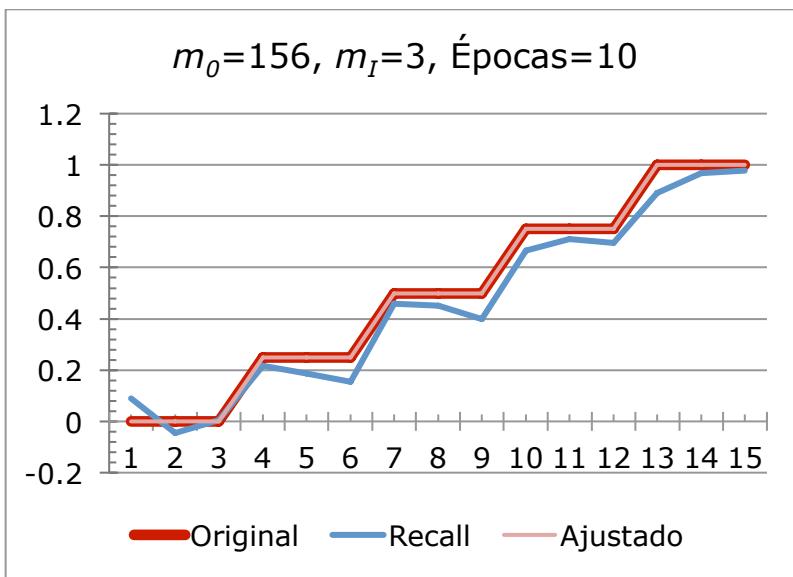


Figura 15. Salida de la clasificación de la imágenes utilizando 156 variables en MLP

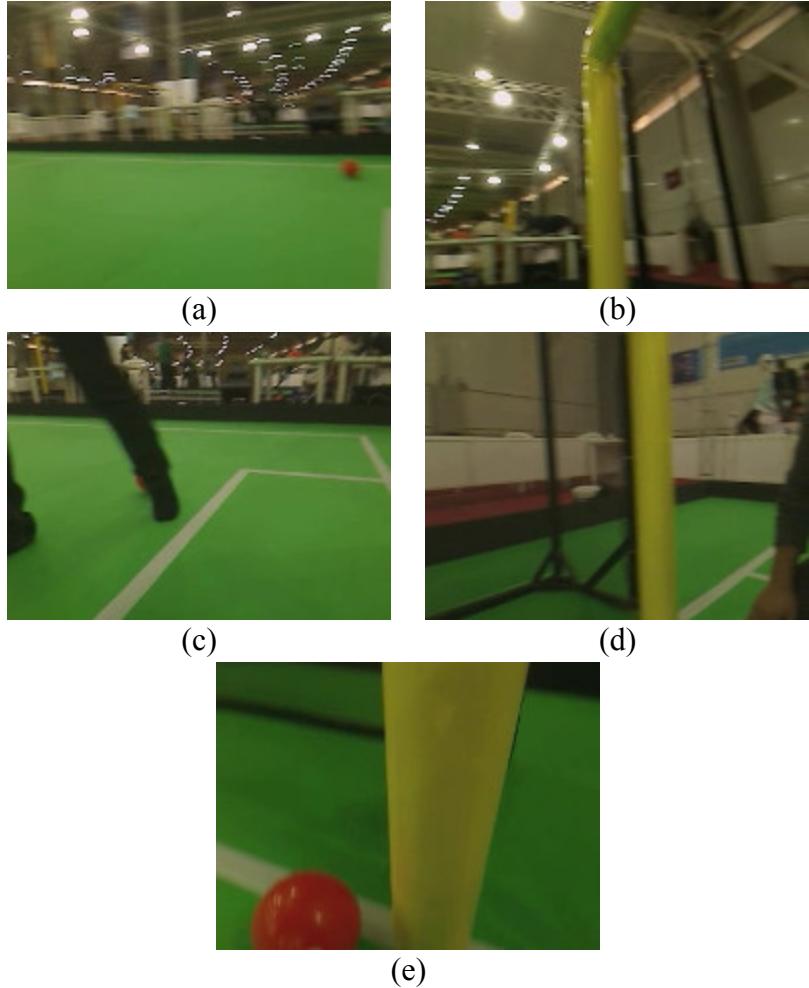


Figura 16. Imágenes utilizadas para la prueba de generalización de la MLP

Probamos ambas redes ya entrenadas con 5 imágenes tomadas del mismo vídeo 13 segundos antes de las imágenes con las que se hicimos el entrenamiento, ver figura 16. Nótese, que las imágenes de la figura 16 son distintas a las imágenes de la figura 15. Por ejemplo la figura 16.a esta borrosa y no muestra una línea que sí esta en la figura 15.a, las figuras 16.b, 16.d muestran un desplazamiento horizontal de un poste respecto a las imágenes 15.b. y 15.d. En la figura 16.e tiene un desplazamiento vertical de la pelota respecto a la figura 15.c, etc. Aún con las diferencias descritas, al utilizar estas imágenes también logramos una clasificación del 100%. Ver figuras 17 y 18.

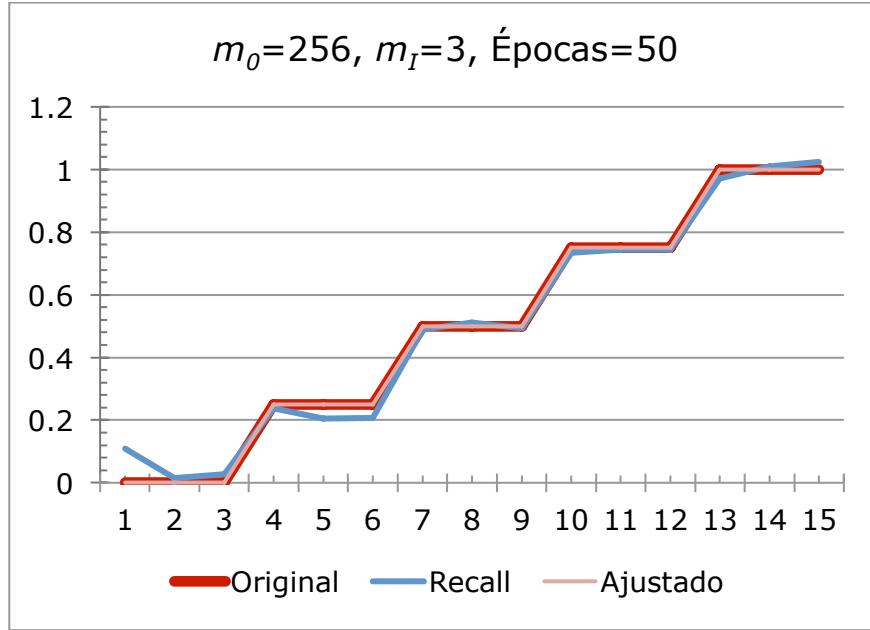


Figura 17. Salida de la clasificación de la imágenes utilizando 256 variables en MLP

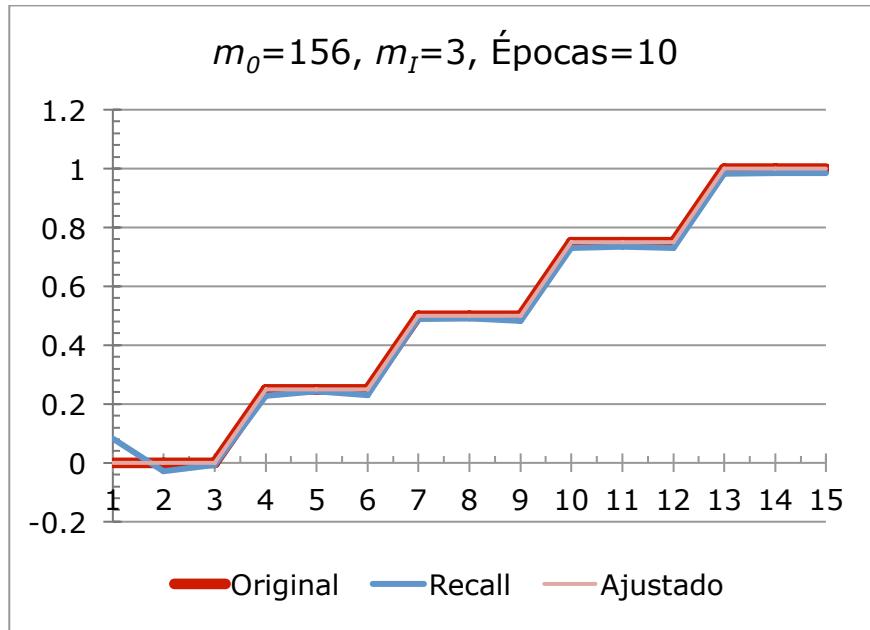


Figura 18. Clasificación de las Imágenes de prueba usando 156 variables en la MLP

De igual forma probamos ambas redes con 5 imágenes tomadas del mismo vídeo 13 segundos después, ver figura 19. En este caso el desplazamiento en las imágenes 19.a, 19.c y 19.e respecto a las correspondientes en la figura 13 es menos evidente. La figura 19.c no tiene una objeto obscuro a la izquierda como en la figura 13.c. En la figuras 19.b se nota un ligero desplazamiento hacia abajo y en la figura 19.d parece ser el poste amarillo esta mas cerca. En esta prueba logramos reconocer exitosamente las imágenes 19.b y 19.c. Ver figuras 20 y 21.

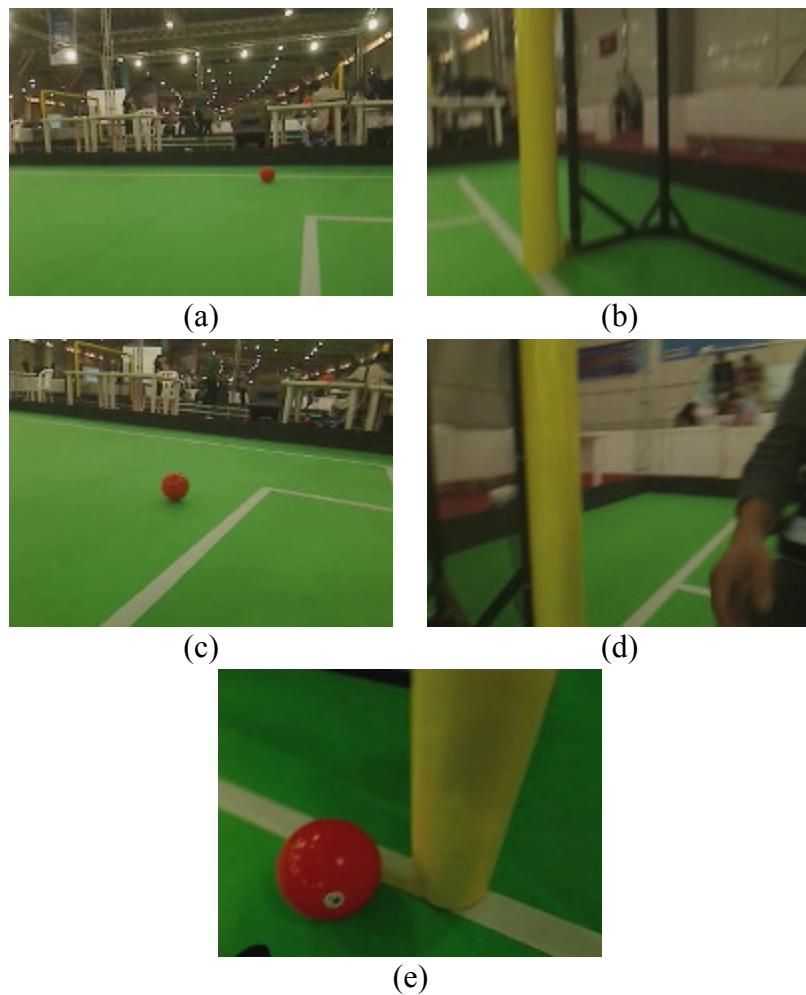


Figura 19. Imágenes utilizadas para la prueba de generalización de la MLP

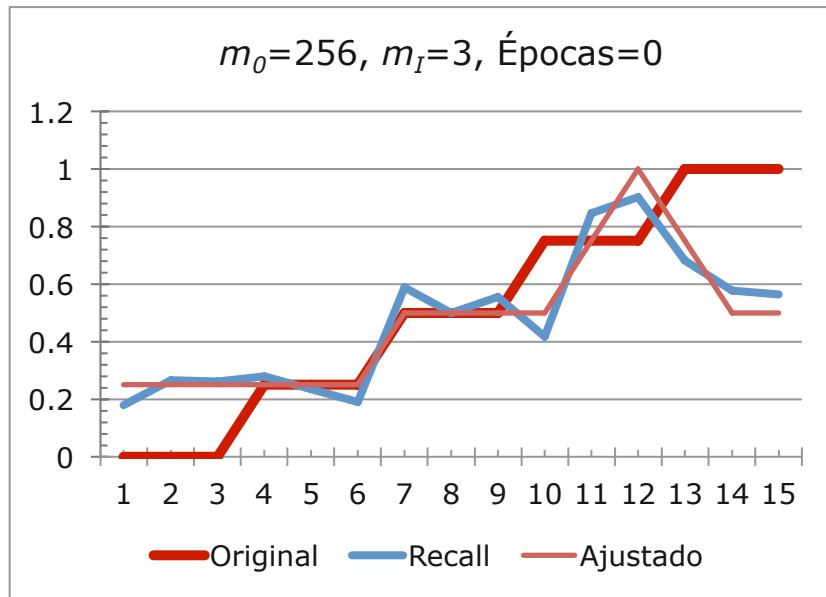


Figura 20. Salida de la segunda clasificación para las imágenes de la Figura 19 utilizando 256 variables en MLP

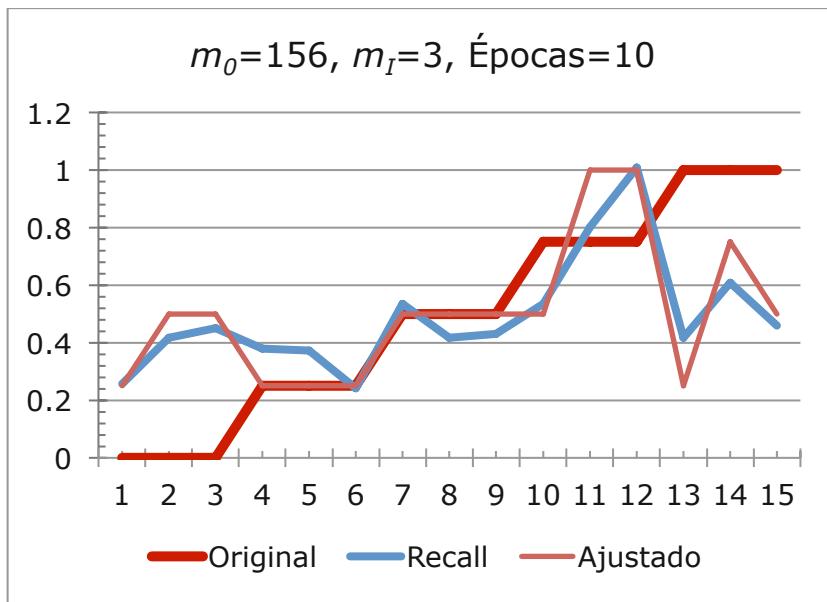


Figura 21. Clasificación de las imágenes de la figura 19 usando 156 variables en la MLP

Estrictamente la metodología que utilizamos en la selección de las imágenes para este experimento es diferente a la planteada durante este trabajo y descrita en la sección 6.2. Ya que no estamos usando vistas previas del ambiente. En su lugar usamos las mismas vistas pero desplazadas en el tiempo. Con esto estamos mostrando lo que pasaría si el robot se desplazara sin conocimiento previo del ambiente. Mientras el robot se desplaza, compara lo que está “viendo” contra lo “vio” t segundos atrás para determinar si está o no en el mismo lugar. Sin embargo, en cierto sentido esto es equivalente.

Estos resultados son alentadores ya que nos dan un indicio que esta metodología funcione para una mayor cantidad de imágenes.

Incluimos 2 anexos donde mostramos con más detalle los resultados que obtuvimos utilizando la norma L_∞ como criterio de dominancia en MOEGA. El anexo 1 contiene cada una de las salidas del MOEGA. En él se observa que en la mayoría de los casos se obtiene un punto solución perteneciente al conjunto de soluciones óptimas de Pareto con pocos individuos y pocas generaciones. En el anexo 3 se muestran los resultados de la comparación de los métodos de aproximación multivariada sobre un conjunto de problemas obtenidos del repositorio de aprendizaje de maquina de UCI [40].

8 Siguientes pasos

1. Hacer una comparación entre los puntos obtenidos por el método del Cedazo y por otro algoritmo (ORB)
2. Aumentar el número de imágenes
3. Simular el comportamiento del robot

Estamos trabajando en un artículo en donde se describa esta metodología. El nombre del artículo es: “*Determination of the position of a mobile robot from the spectrum of the images of a known landscape*”. Lo vamos a enviar a la revista: “*Neural computation and applications*, MIT press, artificial intelligence Springer”. Una segunda opción es

para enviarlo a: “Plos One” -- metodología es lo que cuenta no los resultados checar estas revista

9 Conclusiones y discusión

En este trabajo de investigación desarrollamos una metodología para determinar la posición de un robot. Durante este proceso hemos llegado a varias conclusiones: 1) El mejor método para resolver el problema de multi-objetivo para este trabajo, fue el algoritmo genético MOEGA que utiliza la norma L_∞ para minimizar a) el número de píxeles y b) el error de identificación. 2) El pre-procesamiento de los datos para cada uno de los algoritmos estudiados es fundamental para que funcionen adecuadamente. 3) La mejor forma para conformar el espectro de la imagen es por medio de las frecuencias obtenidas a partir de la información de cada imagen obtenida por el método del Cedazo. Analizamos distintos enfoques de aproximación multivariada: i) Los GMP y ii) las MLP. 5) En (i) el algoritmo FA resultó ser el mejor que LM y PM. Además FA puede utilizar una función no derivable como modelo, lo cual en LM y PM no es posible. Al ser un algoritmo de intercambio utiliza un subconjunto de los datos de entrenamiento para encontrar el modelo de aproximación, a diferencia de LM y PM que requieren de todo el conjunto de datos para solucionar los sistemas de ecuaciones necesarios para encontrar el modelo, por lo que FA requiere de menos memoria que los otros dos. 6) En ii) Determinamos ambos enfoques, a) GMP y b) MLP, son equivalentes y cada uno tiene sus ventajas y desventajas. 7) Por una lado los GMP tienen la ventaja que obtenemos una expresión algebraica que modela el problema de aproximación que queremos resolver. Sin embargo sufren el problema de alta dimensionalidad, es decir para una gran cantidad de variables el número de términos es muy grande. 8) Por otro lado con las MLP se logra una error de aproximación menor. Sin embargo es necesario determinar su arquitectura. La arquitectura de la red la determinamos siguiendo la metodología de [28]. 9) La identificación de la posición la resolvimos utilizamos las MLPs con las que obtuvimos una tasa de reconocimiento del 100% para nuestra base de datos experimental y para un conjunto de imágenes desconocidas.

En el anexo 4 incluimos el código de todos los programas utilizados en este trabajo.

10 Publicaciones a la fecha

- Kuri-Morales, A., López-Peña, I.: Un Nuevo Enfoque a la Optimización Multi-Objetivo con un Algoritmo Genético. Una Aplicación al Sistema de Posicionamiento de un Robot Móvil. Rev. Iztatl Comput. 9, 14–23 (2016).
- Lopez-Peña, I., Kuri-Morales, A.: Multivariate Approximation Methods Using Polynomial Models: A Comparative Study, (2015).

11 Referencias

1. Bay, H. et al.: SURF: Speeded up robust features. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 404–417 (2006).
2. Bloom, C.: PPMZ–High Compression Markov Predictive Coder.
3. Bonin-Font, F. et al.: Visual Navigation for Mobile Robots: A Survey. *J. Intell. Robot. Syst.* 53, 263–296 (2008).
4. Borenstein, J. et al.: Mobile robot positioning: Sensors and techniques. *J. Robot. Syst.* 14, 4, 231–249 (1997).
5. Calonder, M. et al.: BRIEF : Binary Robust Independent Elementary Features. *Eur. Conf. Comput. Vis.* 778–792 (2010).
6. Chao, H. et al.: A survey of optical flow techniques for robotics navigation applications. *J. Intell. Robot. Syst. Theory Appl.* 73, 1–4, 361–372 (2014).
7. Chase, N. et al.: A Benchmark Study of Multi-Objective Optimization Methods Multi-objective Optimization Problem. (2008).
8. Cheney, E.W.: Introduction to Approximation Theory. McGraw-Hill Book Company (1966).
9. Cherkassky, V. et al.: An overview of prediction learning and function approximation. From statistics to neural networks: theory and pattern recognition applications. Springer Science & Business Media (2012).
10. Choset, H.M.: Principles of Robot Motion: Theory, Algorithms, and Implementation. (2005).
11. Cilibarsi, R., Vitányi, P.M.B.: Clustering by compression. *IEEE Trans. Inf. Theory.* 51, 4, 1523–1545 (2005).
12. Coello, C.A.C., Christiansen, A.D.: Two new GA-based methods for multiobjective optimization. *Civ. Eng. Environ. Syst.* 15, 3, 207–243 (1998).
13. Coello, C. a C. et al.: Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition. (2007).
14. Cortez, P. et al.: Modeling wine preferences by data mining from physicochemical properties. *Decis. Support Syst.* 47, 4, 547–553 (2009).
15. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* 2, 4, 303–314.
16. Deb, K. et al.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6, 2, 182–197 (2002).
17. Dellaert, F. et al.: Monte Carlo localization for mobile robots. *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on.* pp. 1322–1328 vol.2 (1999).
18. DeSouza, G.N., Kak, A.C.: Vision for mobile robot navigation: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 2, 237–267 (2002).
19. Gaspar-Cunha, a., Covas, J. a.: RPSGAe - Reduced Pareto Set Genetic Algorithm: A Multiobjective Genetic Algorithm with Elitism. *Work. Mult. Object. Metaheuristics.* (2002).
20. Haykin, S.: Neural Networks: A Comprehensive Foundation. (1998).
21. Horn, B.K.P., Schunck, B.G.: Determining optical flow. *Artif. Intell.* 17, 1–3, 185–

- 203 (1981).
22. Hornik, K.: Approximation capabilities of multilayer feedforward networks. *Neural Networks*. 4, 2, 251–257 (1991).
 23. Hornik, K. et al.: Multilayer Feedforward Networks Are Universal Approximators. *Neural Netw.* 2, 5, 359–366 (1989).
 24. Huang, H., Gartner, G.: A Survey of Mobile Indoor Navigation Systems. *Lect. Notes Geoinf. Cartogr.* 1992, 305–319 (2010).
 25. Hyams, D.G.: CurveExpert Basic, <https://www.curveexpert.net/products/curveexpert-basic/>.
 26. Jianbo Shi, C. Tomasi: Good features to track. Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94. pp. 593–600 (1994).
 27. Kruskal, J.B.: Nonmetric multidimensional scaling: A numerical method. *Psychometrika*. 29, 2, 115–129 (1964).
 28. Kuri-Morales, A.: The Best Neural Network Architecture. 1, 1–13.
 29. Kuri-Morales, A., Aldana-Bobadilla, E.: The Best Genetic Algorithm I. In: Castro, F. et al. (eds.) *Advances in Soft Computing and Its Applications: 12th Mexican International Conference on Artificial Intelligence, MICAI 2013*, Mexico City, Mexico, November 24-30, 2013, Proceedings, Part II. pp. 1–15 Springer Berlin Heidelberg, Berlin, Heidelberg (2013).
 30. Kuri-Morales, A., Cartas-Ayala, A.: Polynomial Multivariate Approximation with Genetic Algorithms. In: Sokolova, M. and van Beek, P. (eds.) *Advances in Artificial Intelligence: 27th Canadian Conference on Artificial Intelligence, Canadian AI 2014*, Montreal, QC, Canada, May 6-9, 2014. Proceedings. pp. 307–312 Springer International Publishing, Cham (2014).
 31. Kuri-morales, A., Lopez-Peña, I.: Experimental validation of an evolutionary method to identify a mobile robot ' s position. (2012).
 32. Kuri-Morales, A., López-Peña, I.: Un Nuevo Enfoque a la Optimización Multi-Objetivo con un Algoritmo Genético. Una Aplicación al Sistema de Posicionamiento de un Robot Móvil. *Rev. Iztatl Comput.* 9, 14–23 (2016).
 33. Kuri-Morales, A., Lopez, J.I.: A Novel Method to Determine a Robot's Position Based on Machine Learning Strategies. 2011 10th Mex. Int. Conf. Artif. Intell. 97–101 (2011).
 34. Kuri-Morales, A.F.: Categorical Encoding with Neural Networks and Genetic Algorithms. *AICT (Applications Comput. Comput. Theory)*. 1, 167–175 (2015).
 35. Kuri-Morales, A.F. et al.: The Best Genetic Algorithm II. In: Castro, F. et al. (eds.) *Advances in Soft Computing and Its Applications: 12th Mexican International Conference on Artificial Intelligence, MICAI 2013*, Mexico City, Mexico, November 24-30, 2013, Proceedings, Part II. pp. 16–29 Springer Berlin Heidelberg, Berlin, Heidelberg (2013).
 36. Kuri-Morales, A.F., Juárez-Almaraz, F.: Genetic multivariate polynomials: An alternative tool to neural networks. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*. 3773 LNCS, 262–270 (2005).
 37. Kuri Morales, A., Galaviz Casas, J.: *Algoritmos genéticos / Angel Kuri, José Galaviz*. Sociedad Mexicana de Inteligencia Artificial, México (2007).
 38. Levenberg, K.: A Method for the Solution of Certain Non-linear Problems in Least-Squares. *Q. Appl. Math.* 2, 2, 164–168 (1944).
 39. Li, M., Vitányi, P.M.B.: An Introduction to Kolmogorov Complexity and Its

- Applications. Springer (1997).
40. Lichman, M.(University of California, Irvine, S. of I. and C.S.: UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml>, (2013).
 41. Lopez-Peña, I., Kuri-Morales, A.: Multivariate Approximation Methods Using Polynomial Models: A Comparative Study. 2015 Fourteenth Mexican International Conference on Artificial Intelligence (MICAI). pp. 131–138 (2015).
 42. Lopez-Peña, J.I.: A METHOD TO DETERMINE A HUMANOID ROBOT POSITION BASED ON MACHINE LEARNING STRATEGIES. Universidad Nacional Autónoma de México (2011).
 43. López-Peña, J.I.: Similarity Measures. A METHOD TO DETERMINE A HUMANOID ROBOT POSITION BASED ON MACHINE LEARNING STRATEGIES. pp. 38–40 (2011).
 44. Lowe, D.G.: Object recognition from local scale-invariant features. Proceedings of the Seventh IEEE International Conference on Computer Vision. pp. 1150–1157 (1999).
 45. Lucas, B.D., Kanade, T.: An Iterative Image Registration Technique with an Application to Stereo Vision. Proceedings of Imaging Understanding Workshop. pp. 121–130 (1981).
 46. Lv, Q. et al.: Multi-probe LSH: efficient indexing for high-dimensional similarity search. Proc. 33rd Int. Conf. Very large data bases. 950–961 (2007).
 47. Madsen, K. et al.: METHODS FOR NON-LINEAR LEAST SQUARES PROBLEMS. (2004).
 48. Mahalanobis, P.C.: On the generalised distance in statistics. Proc. Natl. Inst. Sci. India. 2, 1, 49–55 (1936).
 49. Marquardt, D.W.: An Algorithm for Least-Squares Estimation of Nonlinear Parameters, (1963).
 50. Matthies, L., Shafer, S.A.: Error Modeling in Stereo Navigation. IEEE J. Robot. Autom. 3, 3, 239–248 (1987).
 51. Mitchell, M.: An Introduction to Genetic Algorithms (Complex Adaptive Systems). (1998).
 52. Montibeller, G., Franco, A.: Handbook of Multicriteria Analysis. Handbook of Multicriteria Analysis. pp. 287–310 (2010).
 53. Rosten, E. et al.: Faster and better: a machine learning approach to corner detection. IEEE Trans. Pattern Anal. Mach. Intell. 32, 1, 105–19 (2008).
 54. Rublee, E. et al.: ORB: An efficient alternative to SIFT or SURF. Proc. IEEE Int. Conf. Comput. Vis. 2564–2571 (2011).
 55. Shampine, L.F. et al.: Spline Interpolation. Fundamentals of numerical Computing. pp. 110–112 (1996).
 56. Siegwart, R., Nourbakhsh, I.R.: Introduction to Autonomous Mobile Robots. (2004).
 57. Talukder, A., Matthies, L.: Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. Intell. Robot. Syst. 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ Int. Conf. 4, 3718–3725 (2004).
 58. Teahan, W.: Probability estimation for PPM. Proc. NZCSRSC '95. 1–14 (1995).
 59. Thrun, S.: Probabilistic Algorithms in Robotics. Science (80-.). 4, April, 93–109 (2000).
 60. Villegas Quezada, C., Climent Vilaró, J.: Holistic Face Recognition using Multivariate Approximation , Genetic Algorithms and AdaBoost Classifier :

- Preliminary Results. World Congress on Science, Engineering and Technology. pp. 802–806 , Venice (2008).
61. Weisstein, E.W.: “L^{infinity}-Norm.” <http://mathworld.wolfram.com/L-Infinity-Norm.html>.
 62. Zhou, C.Z.C. et al.: Mobile robot self-localization based on global visual appearance features. 2003 IEEE Int. Conf. Robot. Autom. (Cat. No.03CH37422). 1, 1271–1276 (2003).
 63. Zitzler, E. et al.: Comparison of multiobjective evolutionary algorithms: empirical results. Evol. Comput. 8, 2, 173–195 (2000).
 64. Pearson correlation coefficient,
https://en.wikipedia.org/wiki/Pearson_correlation_coefficient.