

Una nueva medida de compatibilidad para la resolución automática de puzles

David Atienza, José F. Diez-Pastor, César García-Osorio, and Álvaro Arnaiz-González

University of Burgos,
{datienza,jfdpastor,cgosorio,alvarag}@ubu.es

Resumen La resolución de puzles, además de un entretenido pasatiempo, es un reto computacional con importantes aplicaciones en campos como la biología o la arqueología. De todos los tipos de puzle que se pueden resolver, los más complicados son los de piezas cuadradas. En este tipo de puzles, la medida que indica la compatibilidad entre las piezas y guía el proceso solo puede utilizar información relacionada con el color, puesto que todas las piezas tienen la misma forma. Este trabajo presenta una nueva medida de compatibilidad, basada en interpolación bilineal que obtiene resultados competitivos cuando se compara con una de las medidas de referencia en el estado del arte.

Keywords: Pomeranz, puzle, interpolación bilineal

1. Introducción

Los puzles son un entretenido pasatiempo consistente en reconstruir una imagen a partir de fragmentos que no se solapan. Tienen su origen en 1760 cuando el cartógrafo londinense John Spilsbury corta uno de sus mapas en piezas, cada una de ellas correspondiente a cada uno de los países [10]. Por décadas los puzles fueron usados por diversión y para enseñar geografía a los niños, pero a mediados del siglo XX surge una nueva utilidad: resolver puzles como un reto computacional [7].

Desde el punto de vista computacional, el tipo de puzle más complicado es el de piezas cuadradas, dado que la única información disponible para conocer las vecindades de las piezas es la intensidad y el color de estas. Formalmente se trata de un problema NP-completo [5], con un puzle típico de n piezas, podemos combinar las piezas de $n!$ maneras diferentes.

Aparte del aspecto lúdico, es interesante mencionar que el problema de la resolución de puzles tiene también sus aplicaciones en áreas tan dispares como la biología [11], el procesamiento del habla [18], la reconstrucción de objetos arqueológicos [1], la reconstrucción de documentos [4] o la criptografía [6,2].

Las partes de las que se compone un sistema de resolución de puzles son las siguientes: *a)* preprocesado, *b)* obtención de compatibilidad entre piezas, *c)* ensamblado de piezas y *d)* postprocesado. En el preprocesado se obtienen las piezas

que forman el puzle a partir de una imagen inicial tomada por una cámara y en el postprocesado se calculan el desplazamiento y la rotación adecuada para cada pieza para formar la imagen final (para reducir el tiempo de los experimentos, en las comparaciones realizadas en este artículo no se han tenido en cuenta las rotaciones). Estas etapas solo existen cuando se resuelven puzles reales. La mayoría de los artículos recientes generan las piezas del puzle de forma automática a partir de la imagen original, dividiéndola en n piezas [3,14,8,16,12]. De esta manera, se pueden conseguir puzles de cualquier número de piezas y se evitan la fase de preprocesado y postprocesado. El trabajar con imágenes en lugar de puzles de verdad permite la comparación entre algoritmos de distintos autores de manera más fácil.

Posiblemente la etapa más importante en la resolución de puzles es el cálculo de la medida de compatibilidad entre piezas. Una medida de compatibilidad óptima devolvería un 1 para dos bordes que son vecinos y 0 para el resto de casos, en este caso un algoritmo voraz podría resolver el puzle en un tiempo polinomial.

Esta compatibilidad es la única información que guía el proceso de ensamblado y tiene una gran influencia tanto en el tiempo de ejecución como en la calidad de la solución final.

En este artículo se presenta una nueva medida de similitud y se compara con una de las que obtiene mejores resultados hasta la fecha [14]. Para ello se va a evaluar la medida de similitud por sí sola, midiendo su capacidad para encontrar las vecindades correctas de las piezas y se va a evaluar el desempeño de la medida como parte de un sistema de resolución automática de puzles. En este caso se va a usar el algoritmo de Pomeranz et al [14].

Las partes de las que consta este trabajo son las siguientes: La Sección 2 describe el algoritmo de Pomeranz, a continuación, en la Sección 3 se presenta un breve resumen de las medidas de similitud más recientes. La nueva medida de similitud presentada se describe en la Sección 4. La Sección 5 presenta unas breves nociones sobre las distintas medidas de calidad utilizadas para evaluar la calidad de un sistema de resolución de puzles. La Sección 6 describe la validación experimental y presenta los resultados de la comparación entre medidas. Finalmente en la Sección 7 se analizan los resultados y se discuten posibles líneas futuras.

2. Algoritmo de Pomeranz

El algoritmo de Pomeranz ha supuesto un hito en la resolución de puzles, ha conseguido resolver puzles de tamaños que no había sido posible resolver antes de su propuesta, y además la solución del puzle es llevada a cabo sin ningún tipo de pista, conocimiento acerca de la solución o intervención humana de ningún tipo. Solamente utiliza la información proporcionada por la medida de similitud.

El algoritmo de Pomeranz es un algoritmo voraz que consta de tres fases, que se repiten iterativamente hasta obtener la solución final (ver Algoritmo 1).

El algoritmo parte de una pieza inicial, llamada semilla. El bucle principal (líneas 2-6) se ejecuta hasta que se alcanza una solución suficientemente buena. Para determinar como de buena es una solución y guiar el proceso, Pomeraz introduce una métrica de estimación basada en contar el número de «mejores colegas» (*best buddies*). Dos piezas son «mejores colegas» si se reconocen respectivamente como el vecino más probable entre todas las piezas. Este reconocimiento mutuo proporciona mayor seguridad de que son realmente vecinos.

Formalmente, siendo X el conjunto de piezas, R el conjunto de posibles relaciones entre 2 piezas, (*arriba, abajo, derecha e izquierda*) y $C(x_i, x_j, r_k)$ una relación que devuelve la confianza entre dos piezas en la dirección indicada por la relación r_k (por ejemplo, si r_k fuera *arriba*, sería la confianza entre el borde inferior de la pieza x_i y el superior de la pieza x_j), dos piezas x_i y x_j son «mejores colegas» para relaciones opuestas r_1 y r_2 si y solo si: $\forall x_k \in X, C(x_i, x_j, r_1) \geq C(x_i, x_k, r_1)$ y $\forall x_p \in X, C(x_j, x_i, r_2) \geq C(x_j, x_p, r_2)$.

La colocación de piezas (línea 3) sigue las siguientes directrices:

- Se colocan primero las piezas en aquellos huecos en los que haya más vecinos colocados. Este criterio garantiza que el puzle tienda a mantener una forma rectangular. Es un criterio similar al visto en Wolfson *et al* [17], Goldberg *et al* [9] o Nielsen *et al* [12].
- Si de las piezas disponibles para colocar hay alguna que sea «mejor colega» de una ya colocada, se elige aquella que tenga mayor similitud. Si no hay «mejores colegas», se elige simplemente la que tenga mayor similitud con alguna pieza ya colocada.

Después de colocar todas las piezas, se tiene una solución parcial en la que probablemente haya vecinos bien colocados. Sin embargo, es posible que las regiones con vecinos bien colocados estén desplazadas de su posición correcta. Las regiones con vecinos bien colocados se denominan segmentos.

Se determina el mapa de segmentos (línea 4). Para ello se utiliza un algoritmo de segmentación basado en crecimiento de regiones [13], con semillas aleatorias y un predicado que determina que una pieza del puzle pertenece al mismo segmento que su vecino, si son «mejores colegas».

Una vez que se ha obtenido el mapa de segmentos, se calcula el segmento más grande, es decir, el segmento que tiene mayor número de piezas (línea 5).

En la siguiente iteración se vuelve iniciar de nuevo el proceso de colocación de piezas utilizando como semilla el segmento más grande obtenido en la iteración anterior. Se itera hasta que la métrica de «mejores colegas» converja a un máximo local. De este modo, en cada iteración, se obtiene un segmento cada vez más grande, en el caso ideal un segmento que contiene todas las piezas del puzle.

3. Medidas de compatibilidad

En esta sección se presentan algunos conceptos teóricos y se describe un método de calcular la compatibilidad «compatibilidad basada en predicción»

Algorithm 1 Algoritmo resolución de puzles de Pomeranz**Require:** índice de semilla inicial i

```

1:  $Seed = i$ 
2: while  $Rating_{iter} > Rating_{iter-1}$  do
3:    $Puzzle =$  Coloca las piezas a partir de  $Seed$ 
4:    $SegmentMap =$  Obtener el mapa de segmentos de  $Puzzle$ 
5:    $Seed =$  Encontrar el segmento más grande en  $SegmentMap$ 
6: end while
7: return  $puzzle$ 

```

presentada por Pomeranz. La compatibilidad basada en la predicción determina la similitud entre piezas midiendo la distancia entre la predicción de como debería ser el borde de una pieza y el borde de la pieza cuya similitud se pretende comprobar. La Sección 4 presenta una nueva forma de predicción que en la Sección 6 será evaluada experimentalmente con la predicción propuesta por Pomeranz. Las medidas de compatibilidad se calculan sobre filas o columnas (según cual sea la relación de vecindad entre las piezas: arriba, abajo, derecha o izquierda). Una de las medidas de compatibilidad más comunes [3,14,16] es la «compatibilidad basada en disimilitud» (en adelante, disimilitud).

Sean x_j y x_i dos piezas cuadradas de longitud K , se desea calcular la similitud entre el borde izquierdo (última columna) de x_j y el borde derecho (primera columna) de x_i .

La disimilitud es igual a una diferencia de cuadrados y se define como:

$$D(x_i, x_j, r) = \sum_{k=1}^K \sum_{l=1}^3 (x_j(k, K, r) - x_i(k, 1, r))^2 \quad (1)$$

En esta medida se considera que cada píxel contiene tres canales de color diferentes ¹.

Aunque en [3] se encontró que la mejor disimilitud era la suma de cuadrados o norma L_2 , se puede experimentar con nuevas normas $(L_p)^q$.

El valor $D(x_i, x_j, r)$ será bajo en los casos de bordes de piezas muy similares. Varios bordes de piezas que forman parte de una región de la imagen con colores muy uniformes, como por ejemplo el cielo, pueden tener valores bajos de $D(x_i, x_j, r)$ sin ser realmente vecinos. Una compatibilidad óptima debe de ser lo mas discriminatoria posible para evitar este tipo de problemas.

Para atenuar este problema, se normaliza el valor entre 0 y 1 del siguiente modo:

$$C_{i,j,r} = \exp\left(-\frac{D(x_i, x_j, r)}{\text{quartile}(i, r)}\right) \quad (2)$$

Donde $D(x_i, x_j, r)$ es el valor de disimilitud obtenido con la ecuación 1 entre las piezas x_j y x_i cuando la relación entre ellas es que x_j se encuentra a la

¹ En este trabajo las medidas de similitud se calculan usando el modelo de color LAB.

derecha de x_i . Y *quartile* (i, r) es el cuartil de la disimilitud entre x_i y todas las posibles piezas colocadas a la derecha de ésta.

Realizar esta normalización provoca el efecto de que ante dos posibles bordes en las que se haya obtenido la misma $D(x_i, x_j, r)$, se da más valor a aquel en la que haya una mayor diferencia entre el borde más similar y el resto de bordes similares. La normalización también se usa con otros métodos de compatibilidad que no son disimilitud.

La «compatibilidad basada en predicción», se intenta predecir el valor del píxel del borde a partir de los píxels del borde de otra pieza.. Cuanto más se parezca la predicción al valor real, más probabilidades hay de que la pieza sea el vecino correcto. Para calcular la distancia entre la predicción y el valor real se puede utilizar la función euclídea (norma L_2) o las normas $(L_p)^q$. En Pomeranz se describe la «compatibilidad basada en predicción» como:

$$Pred(x_i, x_j, r) = \sum_{k=1}^K \sum_{d=1}^3 \left[\left([2x_i(k, K, d) - x_i(k, K-1, d)] - x_j(k, 1, d) \right)^p + \left([2x_j(k, 1, d) - x_j(k, 2, d)] - x_i(k, K, d) \right)^p \right]^{\frac{q}{p}} \quad (3)$$

En este trabajo los valores usados son $p = 0,3$ y $q = 1/16$.

4. Una nueva medida de compatibilidad

En este trabajo se presenta una nueva medida de compatibilidad: «compatibilidad basada en interpolación bilineal». Se puede ver como un tipo de la «compatibilidad basada en predicción», pero en este caso la predicción del valor de los píxeles se realiza mediante interpolación bilineal en lugar de usando la fórmula 3. La distancia entre la predicción y el valor real se calcula usando una norma $(L_p)^q$ con los mismos valores de p y q descritos anteriormente.

$$\text{diffBilinear}(x_i, x_j, r) = \sum_{k=1}^K \sum_{d=1}^3 \left[\left(x_j(k, K, l) - \text{Bilinear}_j \right)^p + \left(x_i(k, 1, l) - \text{Bilinear}_i \right)^p \right]^{\frac{q}{p}} \quad (4)$$

Siendo x_j y x_i las dos piezas entre las cuales se quiere calcular la compatibilidad. Se puede definir Bilinear_j como una línea de píxeles que se ha obtenido mediante la interpolación bilineal calculada con la penúltima línea de píxeles de x_j y la primera de x_i . Y Bilinear_i la línea se obtiene con la interpolación calculada usando la última línea de x_j y la segunda de x_i .

En la interpolación bilineal cada píxel a predecir ($P_{x,y}$) se obtiene a partir de los cuatro píxeles que lo rodean [15].

5. Métricas de rendimiento

Las métricas de rendimiento para la evaluación de algoritmos de resolución de puzzles fueron introducidas por primera vez por Cho [3] en 2010. Gracias a ellas se puede medir como de bueno es un algoritmo en comparación con otro.

Las métricas introducidas por Cho son las siguientes:

- **Comparación directa:** Calcula el porcentaje de piezas que han sido colocadas en su posición correcta.
- **Comparación de vecinos:** Calcula el porcentaje de vecinos que están correctamente colocados.

La comparación directa es una métrica mucho más pesimista, dado que un puzzle que tenga una o varias regiones de piezas bien colocadas pero ligeramente desplazadas respecto a la solución correcta puede obtener un valor muy bajo con esta medida. La comparación de vecinos es una métrica más fiel y en el caso anterior, se obtendrían valores más altos porque cada pieza tiene los vecinos adecuados.

6. Validación experimental

La siguiente sección describe la validación experimental llevada a cabo para comparar la nueva medida de compatibilidad con la ya existente. Se ha utilizado el repositorio de Cho (20 imágenes formadas por 432 piezas de tamaño 28×28 .) [3].

Se va a comparar la medida propuesta: «compatibilidad basada en interpolación bilineal» contra la «compatibilidad basada en predicción» de Pomeranz et al, la medida que obtiene uno de los mejores resultados del estado del arte.

6.1. Resultados

En la validación experimental, el algoritmo de Pomeranz con la compatibilidad original se ha comparado con la nueva propuesta. Para cada uno de ellos, el algoritmo se ha ejecutado 10 veces con distintas semillas aleatorias iniciales (distintas piezas iniciales) y se ha conservado la solución con mejor valor de mejores colegas. Las figuras 2, 3 y 4 muestra los resultados para cada una de las 20 imágenes en forma de diagrama de barras. El eje x representa el índice de la imagen y el eje y representa el valor de la medida de rendimiento o eficiencia.

En la figura 2 se puede ver el rendimiento del algoritmo de resolución de puzzles de acuerdo a la comparación directa. De las 20 imágenes, la medida de compatibilidad propuesta obtiene 9 victorias (mejores valores de la métrica) por 4 victorias de la de Pomeranz. Además, la medida propuesta consigue resolver el puzzle completamente en 7 de las ocasiones por 5 del rival.

Cuando se mide el rendimiento de acuerdo a la comparación de vecinos (figura 3), la medida propuesta obtiene 8 victorias por 6 del rival.



Figura 1. Repositorio de imágenes usado. Disponible en <http://people.csail.mit.edu/taegsang/Documents/jigsawCode.zip>

En cuanto al número de iteraciones (figura 4), la medida propuesta hace que el algoritmo converja en un número menor de iteraciones para 13 de los casos, empatando en un caso y necesitando más iteraciones solo para 6 imágenes. Este resultado nos hace sospechar que con la nueva medida de compatibilidad, el algoritmo será capaz de resolver puzzles de mayores dimensiones.

7. Conclusiones y líneas futuras

En el presente trabajo se ha presentado una nueva medida de compatibilidad, los experimentos realizados han demostrado que supera a la medida que obtiene uno de los mejores resultados en el estado del arte.

La medida presentada hace converger al algoritmo en un menor número de iteraciones, debido a lo cual puede ser interesante ampliar la validación experimental con nuevos datasets con puzzles más grandes.

Otra línea de mejora es diseñar nuevas medidas de compatibilidad basadas en la interpolación bicúbica, otras formas más avanzadas de interpolación en imágenes o algoritmos de reconstrucción de imágenes o eliminación de ruido. El modo de aplicarlos sería similar a como se usa la interpolación bilineal, usar el algoritmo para predecir el borde de la imagen y posteriormente se convierte en una similitud calculando la distancia con el borde real.

Unos parámetros que aparentan tener una cierta influencia en el desempeño final son los valores p y la q . En la nueva medida se han utilizado los valores optimizados por Pomeranz para su medida. Sería interesante buscar los valores óptimos para la nueva medida o investigar si los valores ideales son dependientes del par de piezas o la imagen a evaluar.

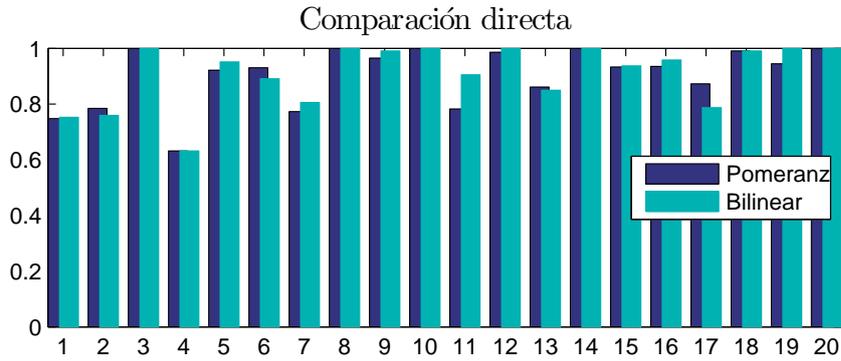


Figura 2. Comparación directa.

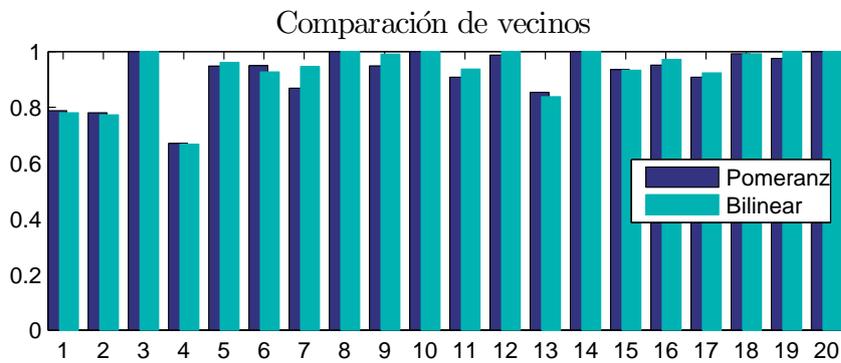


Figura 3. Comparación de vecinos.

Acknowledgments. La investigación realizada en este trabajo ha sido parcialmente subvencionada por el Ministerio de Economía y Competitividad, proyecto TIN-2011-24046.

Referencias

1. Brown, B.J., Toler-Franklin, C., Nehab, D., Burns, M., Dobkin, D., Vlachopoulos, A., Doulas, C., Rusinkiewicz, S., Weyrich, T.: A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Transactions on Graphics (Proc. SIGGRAPH)* 27(3) (Aug 2008)
2. Chang, C.C., Hwang, M.S., Chen, T.S.: A new encryption algorithm for image cryptosystems. *Journal of Systems and Software* 58(2), 83 – 91 (2001), <http://www.sciencedirect.com/science/article/pii/S0164121201000292>
3. Cho, T.S., Avidan, S., Freeman, W.: A probabilistic image jigsaw puzzle solver. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. pp. 183–190 (June 2010)

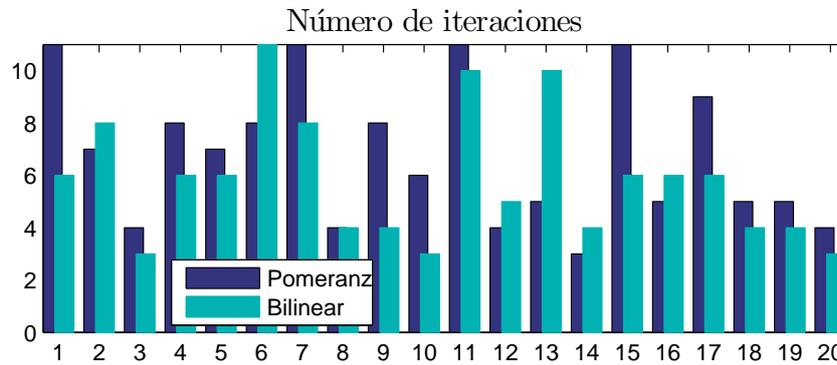


Figura 4. Número de iteraciones

4. Deever, A., Gallagher, A.: Semi-automatic assembly of real cross-cut shredded documents. In: Image Processing (ICIP), 2012 19th IEEE International Conference on. pp. 233–236 (Sept 2012)
5. Demaine, E., Demaine, M.: Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics* 23(1), 195–208 (2007), <http://dx.doi.org/10.1007/s00373-007-0713-4>
6. Farn, E.J., Chen, C.C.: Novel steganographic method based on jig swap puzzle images. *J. Electronic Imaging* 18(1), 013003 (2009), <http://dblp.uni-trier.de/db/journals/jei/jei18.html#FarnC09>
7. Freeman, H., Garder, L.: Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *Electronic Computers, IEEE Transactions on EC-13(2)*, 118–127 (April 1964)
8. Gallagher, A.: Jigsaw puzzles with pieces of unknown orientation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. pp. 382–389 (June 2012)
9. Goldberg, D., Malon, C., Bern, M.: A global approach to automatic solution of jigsaw puzzles. In: Proceedings of the Eighteenth Annual Symposium on Computational Geometry. pp. 82–87. SCG '02, ACM, New York, NY, USA (2002), <http://doi.acm.org/10.1145/513400.513410>
10. Library, T.B.: Jigsaw puzzle, 1766, <http://www.bl.uk/learning/artimages/maphist/minds/jigsawpuzzle/jigsawpuzzle1766.html>, [Internet; Accedido: 6-julio-2014]
11. Marande, W., Burger, G.: Mitochondrial dna as a genomic jigsaw puzzle. *Science* 318(5849), 415 (2007), <http://www.sciencemag.org/content/318/5849/415.abstract>
12. Nielsen, T.R., Drewsen, P., Hansen, K.: Solving jigsaw puzzles using image features. *Pattern Recognition Letters* 29(14), 1924 – 1933 (2008), <http://www.sciencedirect.com/science/article/pii/S0167865508001931>
13. Pitas, I.: Digital Image Processing Algorithms and Applications. AWiley-Interscience publication, Wiley (2000), https://books.google.es/books?id=VQs_Ly4DYDMC
14. Pomeranz, D., Shemesh, M., Ben-Shahar, O.: A fully automated greedy square jigsaw puzzle solver. In: Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on. pp. 9–16 (June 2011)

15. Russ, J.: The Image Processing Handbook, Sixth Edition. CRC Press (2011), <https://books.google.es/books?id=gxXXRJWfEsoC>
16. Sholomon, D., David, O., Netanyahu, N.: A genetic algorithm-based solver for very large jigsaw puzzles. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. pp. 1767–1774 (June 2013)
17. Wolfson, H., Schonberg, E., Kalvin, A., Lamdan, Y.: Solving jigsaw puzzles by computer. *Annals of Operations Research* 12(1), 51–64 (1988), <http://dx.doi.org/10.1007/BF02186360>
18. Zhao, Y.X., Su, M.C., Chou, Z.L., Lee, J.: A puzzle solver and its application in speech descrambling. In: Proceedings of the 2007 Annual Conference on International Conference on Computer Engineering and Applications. pp. 171–176. CEA'07, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA (2007), <http://dl.acm.org/citation.cfm?id=1348258.1348289>