Abstract—In this paper a quantum inspired differential evolution algorithm (QDEA) for image registration is presented. Image registration is a fundamental task in almost every computer vision system. It aims to find the best transformation that allows the superimposing of the common parts of two images. The proposed algorithm is a novel hybridization between differential evolution algorithms and quantum computing. So, differential evolution algorithms have been enhanced by the use of quantum concepts such as quantum bit and states superposition. Mutual information has been used as a similarity measure. The obtained results are very promising. Our algorithm has provided a fast convergence and optimal registration quality.

Keywords—Quantum computing, differential evolution, image registration, mutual information.

I. INTRODUCTION

Differential evolution algorithms (DEAs) are population based algorithms like genetic algorithms using similar operators; crossover, mutation and selection. The main difference in constructing better solutions is that genetic algorithms give more importance to the crossover operation while DEAs consider the mutation operation as the most important one. This main operation is based on the differences of randomly sampled pairs of solutions in the population [1]. Like in evolution strategies, the differential evolution is oriented to optimize real values, but it is very simple to be implemented [2].

Image registration is a central task in most vision systems. It is required in different applications such as objects recognition, 3D reconstruction and data fusion. Basically, the image registration (or image alignment) can be defined as the process which consists of finding the best geometric transformation that allows the superimposing of the common parts of two images. To solve this problem, which is a combinatorial optimization one, many approaches have been proposed. All of them try to reduce the computing complexity and at the same time avoid local optiums. Among the proposed methods we can mention those based on artificial neural networks, simulated annealing, taboo search, genetic algorithms [3], ants colonies, and artificial immune systems.

Another search field called Quantum Computing (QC) has appeared and induced intense researches in the last decade. This evolution that takes its origins from the quantum physics principles reduces remarkably the algorithm's complexity. This is offered by the possibility of parallel computing. Such possibility of parallel computing can be exploited to solve combinatorial optimization problems which use a great set of data. So quantum computing allows the possibility of designing very powerful algorithms. However, these algorithms may not be well exploited before developing of powerful quantum machines. Awaiting the construction of such machines, the idea of simulating quantum algorithms on classical computers or to combine them to other conventional methods has appeared [4]. Within this direction, we are interesting in studying the differential evolution quantum hybridization and its contribution in solving combinatorial optimization problems. Researches in the field of combination between evolutionary algorithms and quantum computing have started at the end of 90’s. The purpose of this combination is to enhance the profit of each one of these two approaches by mutually inspiring each from other.

In this paper we propose a quantum inspired differential evolution algorithm for rigid image registration. The same problem has been tackled in a previous work using a quantum-inspired genetic algorithm [5]. The present work represents an enhancement of [5] by integrating the differential evolution aspect. Consequently, the article will be organized as follows. The section 2 gives some concepts about differential evolution, image registration, mutual information and quantum computing. The proposed approach is described in section 3. Section 4 illustrates some experimental results. Finally, conclusion and some perspectives are drawn up.

II. BASIC CONCEPTS

A. Differential evolution

Differential Evolution (DE) is a novel parallel direct search method which uses N parameter vectors \( \mathbf{x}_{i,G} \), \( i=0..N-1 \) as a population for each generation \( G \). N does not change during the minimization process.

The crucial idea behind DE is a new scheme for generating trial parameter vectors. DE generates new parameter vectors by adding the weighted difference vector between two population members to a third member. If the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with which it was compared. The comparison vector can, but does not need to be a part of the generation process mentioned above.

Differential Evolution algorithms comprise a class of evolutionary algorithms recently proposed in the literature [6,7]. Just like Evolution Strategies they use chromosomes based on floating-point numbers to represent candidate solutions. Each individual is defined as a d-dimensional vector \( \mathbf{a} \in \mathbb{R}^d \) and the whole population of potential solutions at generation \( g \), is given by \( P(g)=\{\mathbf{a}_1,\ldots,\mathbf{a}_\mu\} \).

Here the population size \( \mu \) does not change during the search. The main evolutionary operator in DE is meaningfully different from other evolutionary algorithms since mutation is neither based on the alteration of genes by using a mutation probability nor rest on the use of a defined probability distribution function. In DE the mutation operator mutates \( \mu \) vectors through the weighted difference
of two (or four) other vectors according to:
\[ \mathbf{v}_i = \mathbf{a}_i + F \times (\mathbf{a}_{i-2} - \mathbf{a}_{i-3}) \] (1)
where \( i = 1, 2, ..., \mu \), and the random indexes \( r1, r2, r3 \in [1, 2, ..., \mu] \) are mutually different and also distinct from the index \( i \). \( F \in [0, 2] \) is a real constant which affects the differential variation between two vectors. As in other evolutionary algorithms, the crossover operator is introduced in order to increase the diversity of the population. A differential evolution algorithm can be presented as follow.

```plaintext
Generate random solutions that cover the given space.
Evaluate each solution.
g=1
While (convergence is not reached)
   For i=1 to Population size
      Apply differential mutation.
      Execute differential crossover
      Clip the solutions if necessary
      Evaluate the new solution.
      Apply differential selection.
   End
   g=g+1;
End
```

B. Quantum computing

Twenty years ago, Richard Feynman observed that classical computers could not simulate certain quantum mechanical effects such as entanglement [8]. This observation spawned interest in the field of quantum computing, i.e., computational machines that perform calculations by emulating quantum mechanical effects. This speculation proved justified in 1994 when Peter Shor described a polynomial time quantum algorithm for factoring numbers. In this section we review some of the basic concepts of quantum computing.

Quantum computing is based upon physical principles from the theory of quantum mechanics, which is in many ways counterintuitive. Classical computer systems represent a single bit of information deterministically: the value is either logic 0 or logic 1. Quantum computer systems represent a single bit of information as a quantum bit (qubit), which is a unit vector in the complex Hilbert space \( \mathbb{C}^2 \).

- **Quantum bit (qubit)**
  - Using the bra/ket notation introduced by Dirac[9] the qubit state can be given by:
    \[ |\Psi\rangle = \alpha |0\rangle + \beta |1\rangle \] (2)
  - The ket symbol is denoted by \( |\psi\rangle \) and the corresponding bra is denoted by \( \langle \psi| \). The ket describes a quantum state and the corresponding bra is its complex conjugate. \( |0\rangle \) and \( |1\rangle \) represent the classical bit values 0 and 1 respectively; \( \alpha \) and \( \beta \) are complex numbers such that:
    \[ |\alpha|^2 + |\beta|^2 = 1 \] (3)
  - If a superposition is measured with respect to the basis \( \{ |0\rangle, |1\rangle \} \), the probability to measure \( |0\rangle \) is \( |\alpha|^2 \) and the probability to measure \( |1\rangle \) is \( |\beta|^2 \).

- **Quantum register**
  - The qubit register can be prepared in a 3 superposition of all possible classical states \( i \):
    \[ |\text{register}\rangle = \sum_{i=0}^{2^\mu} c_i |i\rangle \] (4)
  - This register satisfies the following equation:
    \[ \sum_{i=0}^{2^\mu} |c_i|^2 = 1 \] (5)
  - Where \( c_i \) is the probability to obtain the state \( i \).

In classical computation, the possible states of a system of \( n \) bits form a vector space of \( n \) dimensions, i.e. we have \( 2^n \) possible states. However, in a quantum system of \( n \) qubits the resulting state space has \( 2^n \) dimensions. It is this exponential growth of the state space with the number of particles that suggests a possible exponential speed-up of computation on quantum computers over classical computers. The basis of the state space of a quantum system of \( n \) qubits is: \( \{|00...0\rangle, |00...1\rangle...|11...1\rangle\} \).

The measurement of a single qubit projects the quantum state onto one of the basis states associated with the measuring device. The result of a measurement is probabilistic and the process of measurement changes the state to that measured. Multi-qubit measurement can be treated as a series of single-qubit measurements in the standard basis.

C. Image registration

Image registration is the process of aligning two or more images of the same scene. Typically, one image, called the base image, is considered the reference to which the other images, called input images, are compared. The object of image registration is to bring the input image into registration with the base image by applying a spatial transformation to the input image. A spatial transformation maps locations in one image to new locations in another image. Image registration is often used as a preliminary step in other image processing applications. For example, we can use image registration to align satellite images of the earth's surface or images created by different medical diagnostic modalities (MRI and SPECT). After registration, we can compare features in the images to see how a river has migrated, how an area is flooded, or to see if a tumor is visible in an MRI or SPECT image.

D. Rigid and Non-Rigid image registration

Generally, to align images, transformations can be divided into rigid and non-rigid transformations. The rigid methods can be used to cope with rotation and translation differences between the aligned images. Specific problems arise due to differences in image sensing for different studies, so elastic or non-rigid methods are required to cope with local differences between the images. The elastic (Non-rigid) transformations are based on models from elasticity theory and describe local deformations. The central idea behind elastic registration is to consider images as continuous bodies and to model the geometric differences between images such that they have been caused by elastic deformation.

- **Intensity based methods for image registration**
  - Registration techniques are very numerous, and are generally classified according to the kind of features they use in the images to deform them. On one hand, intensity-based algorithms use the intensity of the images. On the
other hand, geometric algorithms use geometric features segmented from the images, such as object boundaries. Intensity-based approaches for elastic registration directly exploit the image intensities. The main advantage of these schemes is that an explicit segmentation of the images is not required.

E. Entropy based measures and mutual information

The entropy is a statistic defined by Shannon in 1948 that summarizes the randomness of a given random variable. The more random a variable is, the larger entropy it will have.

Given a random variable represented by a probability distribution $X$, i.e. a set of couples $(x_i, p_i)$ where $p_i$ is the probability of occurrence of the value $x_i$. The entropy of $X$ is given by:

$$H(X) = -\sum p_i \log p_i$$  \hspace{1cm} (6)

Intuitively, entropy measures the average information provided by a given distribution. When dealing with two random variables represented by two probability distributions $X$ and $Y$, we are interested by answering the question: "How likely the two distributions are functionally dependant?" In total dependence case, a measurement of one distribution removes any randomness about the other. As a consequence, quantifying the independence is equivalent to quantifying the randomness. The joint entropy is given by:

$$H(X,Y) = -\sum \sum p(x,y) \log p(x,y)$$  \hspace{1cm} (7)

The mutual information is a measure of the reduction on the entropy of $Y$ given $X$ and is then given by:

$$MI(X,Y)=H(X)+H(Y)-H(X,Y)$$  \hspace{1cm} (8)

The mutual information is maximized when the two variables are totally dependant [3] [5].

III. THE PROPOSED ALGORITHM

Having two images $I_1$ and $I_2$ obtained from either similar or different sensors, the proposed algorithm allows the estimation of the geometric transformation which superposes the two images.

As in differential evolution algorithms, initial solutions are encoded in $N$ chromosomes representing the initial population. The difference in our algorithm is that each chromosome does not encode only one solution but all the possible solutions by putting them within a superposition.

The geometric transformation that aligns the image $I_2$ to $I_1$ is rigid, i.e. a combination of a rotation and a translation. A chromosome encodes the three parameters which are the rotation angle $\theta$ and the translation parameters $dx$ and $dy$. Having such parameters, the position of each pixel in the resulting image $(x', y')$ can be calculated from the original position $(x, y)$ as follows:

$$x' = \cos(\theta) \cdot x - \sin(\theta) \cdot y + dx$$
$$y' = \sin(\theta) \cdot x + \cos(\theta) \cdot y + dy$$  \hspace{1cm} (9)

Each parameter is encoded using a binary representation. A bit in a chromosome does not represent only the value 0 or 1 but a superposition of the two. In this way, all the possible solutions are represented in a chromosome and only one solution among them can be measured at each time according to the probabilities $|\alpha|^2$ and $|\beta|^2$. A chromosome is then represented by:

$$\begin{pmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_L \\ \beta_1 & \beta_2 & \ldots & \beta_L \end{pmatrix}$$

Where, each column represents a single qubit. In our algorithm $\alpha_i$ and $\beta_i$ are real values only.

Initially we generate randomly 3 chromosomes. Each one is composed of $N=24$ qubits, 8 qubits for each parameter:

- $-127 \leq dx \leq +127$;
- $-127 \leq dy \leq +127$;
- $0 \leq \theta \leq 2 \pi$ (the interval is subdivided into $2^8 = 256$ different values)

$dx$ and $dy$ are the 2D translation parameters and belong to the interval and $\theta$ is the rotation angle.

During the whole process we keep in memory the global best solution. The algorithm consists on applying cyclically 3 quantum differential evolution operations:

The first operation is a quantum interference which allows a shift of each qubit in the direction of the corresponding bit value in the best solution. That is performed by applying a unitary quantum operator which achieves a rotation whose angle is function of $\alpha_i$, $\beta_i$ and the value of the corresponding bit in the best solution.

$\delta \theta$ is chosen equal to $\pi/120$. The following table gives the value of the rotation angle in function of $\alpha$, $\beta$ and the bit's value in the best solution.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>Reference bit value</th>
<th>Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>$&gt; 0$</td>
<td>$&gt; 0$</td>
<td>1</td>
<td>$+\theta$</td>
</tr>
<tr>
<td>$&gt; 0$</td>
<td>$&gt; 0$</td>
<td>0</td>
<td>$-\theta$</td>
</tr>
<tr>
<td>$&gt; 0$</td>
<td>$&lt; 0$</td>
<td>0</td>
<td>$+\theta$</td>
</tr>
<tr>
<td>$&lt; 0$</td>
<td>$&gt; 0$</td>
<td>1</td>
<td>$-\theta$</td>
</tr>
<tr>
<td>$&lt; 0$</td>
<td>$&gt; 0$</td>
<td>0</td>
<td>$+\theta$</td>
</tr>
<tr>
<td>$&lt; 0$</td>
<td>$&lt; 0$</td>
<td>1</td>
<td>$+\theta$</td>
</tr>
</tbody>
</table>

The second operation consists of a quantum differential mutation which will perturb values of quantum chromosomes using a difference between two other quantum chromosomes. An example is given below:

Finally, we perform a selection of the chromosomes having the best fitness value. For this, we apply first a measurement on each chromosome to have from it one solution among all those present in superposition. But unlike pure quantum systems, the measurement here does not destroy the states' superposition. Since our algorithm operates on conventional computer and does not require the presence of a quantum machine, it is possible and in our interest to keep all the possible solutions in the superposition for the next iterations. For each measurement result, we extract the transformation parameters and use them to transform the second image and have a calculated image. To evaluate the quality of a solution, we compute the mutual information between the first image (reference image) and the calculated image. Greater the mutual information is, better the solution will be considered.
Afterwards, we select the 2 chromosomes from which derive the 2 best results and we generate also randomly a new chromosome (to maintain a good diversity). So we have all in all 3 chromosomes which form the new population. The global best solution is then updated if a better one is found and the whole process is repeated until we have an acceptable solution.

| 1.0000 | 0.2442 | 0.6121 | 0.5644 | 0.9219 |
| 0.8343 | 0.2574 | 0.6174 | 0.7216 | 0.8913 |
| 0.5514 | 0.9663 | 0.7866 | 0.6923 | 0.4534 |
| 0.4703 | 0.2028 | 0.1987 | 0.8394 | 0.3333 |
| 0.8825 | 0.9792 | 0.9801 | 0.5435 | 0.9428 |
| 0.6361 | 0.2159 | 0.2040 | 0.9966 | 0.3639 |
| 0.7716 | 0.9764 | 0.9790 | 0.9824 | 0.9314 |

Figure 3. Rigid registration for mono-modal images.

Figure 4. Rigid registration for multi-modal images.

The following figure represents the global best solution (on the whole process) and the local best solution (on each generation). Also this diagram shows the impact of the interference on the local solutions.

V. Conclusion

In this article we have presented a quantum-inspired differential evolution algorithm for image registration. This QDEA is based on the maximization of the mutual information between the reference and result images. The use of the mutual information allows the registration of images coming from different sensors. According to the obtained results this algorithm assures convergence and offers very good solutions. We have used the algorithm for optimizing rigid parameters but we can extend to other types of spatial transformations.

This algorithm, compared to genetic algorithms or classical differential evolution algorithms provides the advantage of giving a greater diversity by using quantum coding of solutions, i.e. all the solutions exist within each chromosome and what change are the probabilities to have one of them as a result of a measurement. Therefore, the size of the population does not need to be great. The quantum representation offers great diversity. Because of this, we have not used the cross-over having the role of giving diversity in a classical evolutionary algorithm. Another advantage is that the interference provides in some way a guide for the population individuals and reinforces therefore the algorithm convergence. The proposed QDEA has the advantage of using a small population size and the number of necessary iterations to have acceptable solution is significantly smaller (generally about 100 iterations). The obtained results lead us to think to extend it to solve other important optimization problems.

References