Image Matting via LLE/iLLE Manifold Learning

David Tien, Member, IEEE, Junbin Gao, and Jim Tulip

Abstract—Accurately extracting foreground objects is the problem of isolating the foreground in images and video, called image matting which has wide applications in digital photography. This problem is severely ill-posed in the sense that, at each pixel, one must estimate the foreground and background pixels and the so-called alpha value from only pixel information. The most recent work in natural image matting rely on local smoothness assumptions about foreground and background colours on which a cost function has been established. In this paper, we propose an extension to the class of affinity based matting techniques by incorporating local manifold structural information to produce both a smoother matte based on the so-called improved Locally Linear Embedding. We illustrate our new algorithm using the standard benchmark images and very comparable results have been obtained.

Index Terms—Image Matting, Locally Linear Embedding, Manifold Learning, Alpha Matte, Nesterov’s Method

I. INTRODUCTION

Image matting is the problem of isolating the foreground in a single image. These kind of problems are important in both computer vision and graphics applications as well image/video editing. There is a large amount of research interest in the field and a variety of techniques that have been developed to solve this problem. We refer readers to a 2007 survey article [1] in which a comprehensive review of existing image and video matting algorithms and systems was presented with an emphasis on the advanced techniques that have been recently proposed.

Most common techniques rely on the so-called alpha matte model. The model specifies the image composition process from alpha matte, mathematically defined as follows,

\[ I_i = \alpha_i F_i + (1 - \alpha_i) B_i \]  

where \( I_i \), \( \alpha_i \), \( F_i \) and \( B_i \) are image colour value, the alpha matte value, the foreground image colour value and the background colour value, respectively, at a given pixel \( i \). We assume that the alpha matte value \( \alpha_i \) lies in the range between 0 (for true background pixels) and 1 (for true foreground pixels). This kind of alpha matte are called soft matte, while the hard matte values take either 0 or 1 to clearly distinguish foreground and background pixels.

Matting requires that the \( \alpha_i \), \( F_i \) and \( B_i \) are solved simultaneously with the only available image information \( I_i \). Obviously this is a severely ill-posed problem. In practice, matting requires user interaction to produce acceptable results. One of common approaches relies on the user supplying a trimap, which is an input image which labels some pixels as definite foreground, definite background and some unknown regions. Usually the unknown regions are around the boundary of the objects to be extracted. To infer the alpha matte value in the regions, Bayesian matting [2] uses a progressively moving window marching inward from the known regions. Among the propagation-based approaches, the Poisson matting algorithm [4] assumes the foreground and background colours are smooth in a narrow band of unknown pixels, then solves a homogenous Laplacian matrix; A similar algorithm is proposed in [5] based on Random Walks.

In the closed-form matting approach [6], Levin et al. exploited learning approach to fit a linear model for foreground and background colours in each local window, thus globally a matting Laplacian matrix can be formulated to impose a natural smoothness constraint and finally a quadratic cost function over alpha matte can be minimized to solve the matting problem. As minimizing a quadratic cost function is equivalent to solving a linear system, a closed-form solution can be achieved. One of benefits offered by this approach is that user input can be reduced to several scribbles because of smoothness specified by the Laplacian matrix in the technique. When the image size is large, the resulting linear system is huge and it is time-consuming to solve. In a recent work [7], He et al. proposed a fast matting scheme by using large kernel matting Laplacian matrices. Although the closed-form matting takes care of global smoothness of entire matte, it pays less attention to edge discontinuity. In our recent research we introduced zero-one penalty [8] and total variation regularisation [9] to enforce edge preservation. This has added better improvement over the closed-form matting and other similar learning-based approaches.

A similar approach [10] has been proposed by making use of the Local Tangent Space Alignment (LTSA) [11]. LTSA, as a manifold learning approach, is an unsupervised learning algorithm that computes low-dimensional, neighbourhood-preserving embeddings of high-dimensional inputs. The argument used in [10] is to apply the LTSA algorithm to capture local linear structure among the colour information over a local window and link that to the matting value space. This kind of observation is intuitively vert encouraging for manifold learning methods to be successful in image matting.

In this paper, we consider using traditional manifold learning algorithms [12], [13] such as Locally Linear Embedding (LLE) [14] and its improved version iLLE [15] to solve the
problem of image matting. As has been done in [6], we still use
the so-called user-specified constraints (such as scribbles or a
bounding rectangle). Similar approaches can also be found in
[6], [16], [17]. In deriving the matting Laplacian matrices [6],
the authors have used the assumption that the foreground (or
background) colours in a local window lie on a single line in
the RBG colour space. This assumption is going to be naturally
replaced with the smooth manifold assumption which is purely
a mathematical assumption in learning a smooth manifold.
Hence, a locally linear learning model such as LLE can be
utilized to capture such linear information which in turn assists
mate information learning. One of innovations offered by our
approach is to take a natural definition of pixel neighborhood
on the colour manifold as the neighborhood used in most
standard manifold learning algorithms such as LLE. It is hoped
that the idea of using manifold learning or a dimensionality
reduction approach for image matting as offered in this paper
will inspire more exploration in this direction.

The paper is organized as follows. Section II simply in-
truduces the Locally Linear Embedding (LLE) [13], [14]
and iLLE [15] manifold learning algorithms. Section III is
dedicated to formulating the new algorithms based on LLE.
In Section IV, we present several examples of using the new
image matting approach over the benchmark images, see [6].
We make our conclusions in Section V with suggestions for
further exploring the application of dimensionality reduction
algorithms in image matting.

II. LLE/iLLE ALIGNMENT MATRICES

A. LLE and Its Alignment Matrix

The locally linear embedding (LLE) algorithm has been
recently proposed as a powerful eigenvector method for the
problem of nonlinear dimensionality reduction [14]. In the last
decade, many LLE based algorithms have been developed and
introduced in machine learning community; kernelized LLE
(KLLE) [18], Laplacian Eigenmap (LEM) [19], Hessian LLE
(HLLE) [20], robust LLE [21], weighted LLE [22] enhanced
LLE [23] and supervised LLE [24].

By exploiting the local symmetries of linear reconstruc-
tions, LLE is able to learn the global structure of nonlinear
manifolds, such as those generated by images of faces, or
documents of text. The LLE method is also widely used for
data visualization [14], classification [22], [24] and fault
detection [25].

Because of the assumption that local patches are linear, that
is, each of them can be approximated by a linear hyperplane,
each data point can be represented by a weighted linear
combination of its nearest neighbours (different ways can
be used in defining the neighbourhood). Coefficients of this
approximation characterize local geometries in a highdimen-
sional space, and they are then used to find low-dimensional
embeddings preserving the geometries in a low-dimensional
space. The main point in replacing the nonlinear manifold
with the linear hyperplanes is that this operation does not
bring significant error, because, when locally analyzed, the
curvature of the manifold is not large, i.e., the manifold can
be considered to be locally flat. The result of LLE is a single
global coordinate system.

The argument that we take for using the LLE for image
matting is that, like the application of the Laplacian alignment
matrix used in [6], local linear structure among the colour in-
formation over a local window can be learnt and transferred to
the matting value space. This observation suggests intuitively
that the LLE alignment matrix could be successful in image
matting.

In the following sections, we still use $X_i = \{I_i | ij \in
w_i, j = 1, ..., K\}$ to denote the subset of colour vectors over
a local window pixels of pixel $i$. Note that the pixel $I_i$ is
contained in $X_i$. Under the LLE assumption, the colour vector
$\hat{I}_i$ at pixel $i$ can be approximated by a linear combination
$\beta_{ij}$ (the so-called reconstruction weights) of its $K - 1$ nearest
neighbours $X_i \setminus I_i$. Hence, LLE fits a hyperplane through
$\hat{I}_i$ and its nearest neighbors in the colour manifold defined
over the image pixels. The fitting is achieved by solving the
following optimal problem

$$\min_{W} \sum_{i} \|I_i - \sum_{j=1}^{K} \beta_{ij} I_{ij}\|^2$$

under the condition $\sum_{j=1}^{K} \beta_{ij} = 1$. For the sake of simplicity,
we assume that $\beta_{ij} = 0$ when $I_{ij} = I_i$. Assuming that the
manifold is locally linear, the local linearity may be preserved
in the space of matting values. Once the weights $W$ have
been determined, the matting values $\alpha$ can be determined by
minimizing the following objective function

$$\min_{\alpha} F(\alpha) = \sum_{i} \|\alpha_i - \sum_{j=1}^{K} \beta_{ij} \alpha_{ij}\|^2 = \alpha^T R_{\text{LLE}} \alpha$$

where $R_{\text{LLE}} = (E - W)^T (E - W)$ is called LLE alignment
matrix and $E$ is the identity matrix.

In the standard LLE algorithm, (4) is usually solved with ad-
tional constrained conditions like $\|\alpha\|_2 = 1$ which results in
an eigenvector problem. Instead of such standard constraints,
in image matting we would like to formulate a matting solution
by including appropriate constraints. We will discuss this issue
in the next section.

B. iLLE and Its Alignment Matrix

Recently Xiang et al [15] re-explained the LLE as well as
LTS (Local Tangent Space Alignment) [11] under the
framework of local linear transformation, and a close link
between LLE and LTS has been established. Based on this,
Xiang et al. [15] proposed a new model of improved LLE
(iLLE).

We still use $X_i = \{I_i | ij \in w_i, j = 1, ..., K\}$ to denote the
subset of colour vectors over a local window of pixel $i$. Note
that the pixel $I_i$ is contained in $X_i$. In LLE, a linear regression
was established to regress $\hat{I}_i$ by the rest of pixels $X_i \setminus I_i$ in $X_i$.
In iLLE, we can construct $K$ linear regressions. That is, we
can linearly approximate each pixel $I_{ij}$ with the rest of pixels
in $X_i$. Specifically, we have
\[ I_{ij} = \sum_{l \neq j} \beta_{i,j}^{(l)} I_{il} + \epsilon_{ij}, \quad j = 1, 2, \ldots, K \]

Denote \( \beta_{i,j} = [\beta_{i,j}^{(1)}, \ldots, \beta_{i,j}^{(j-1)}, \beta_{i,j}^{(j+1)}, \ldots, \beta_{i,j}^{(K)}]^T \) a \((K-1)\)-dimensional vector. Then the weight vector $\beta_{i,j}$ can be solved for each $j$ respectively by
\[ \min_{\beta_{i,j}} ||I_{ij} - \sum_{l \neq j} \beta_{i,j}^{(l)} I_{il}||^2 + \lambda ||\beta_{i,j}||^2 \]
under the condition $\sum_{l \neq j} \beta_{i,j}^{(l)} = 1$. A closed-form solution for the weight vector $\beta_{i,j}$ can be given, please refer to [15] for more details. For each $j = 1, \ldots, K$, let define an extended $K$-dimensional vector by $\beta_j^{(i)} = [-\beta_{i,j}^{(1)}, \ldots, -\beta_{i,j}^{(j-1)}, 1, -\beta_{i,j}^{(j+1)}, \ldots, -\beta_{i,j}^{(K)}]^T$, i.e., we insert 1 at the location $j$ in $\beta_{i,j}$ and negate all its components. Then define $K$ small matrices by
\[ C_j^{(i)} = \beta_j^{(i)} \beta_j^{(i) T}, \quad j = 1, 2, \ldots, K \]

Now consider the matte value space, i.e., all the matte values $\alpha = [\alpha_1, \ldots, \alpha_N]^T$ over all $N$ pixels. In the local window $w_i$ at pixel $i$, we wish to find the suitable matte value $\alpha_i = [\alpha_{i1}, \ldots, \alpha_{iK}]^T$ such that under the learned locally linear structures the following error is as small as possible
\[ \sum_{j=1}^{K} ||\alpha_{ij} - \sum_{l \neq j} \beta_{i,j}^{(l)} \alpha_{il}||^2 = \alpha_i C_i \alpha_i^T \]
(3)
where $C_i = \sum_{j=1}^{K} C_j^{(i)}$ is a $K \times K$ matrix.

Optimization (3) can be further extended to the entire matte value space. This is to ask
\[ \min_{\alpha} F(\alpha) = \sum_i \alpha_i C_i \alpha_i^T = \alpha^T R_{\text{iLLE}} \alpha \]
(4)
where $R_{\text{iLLE}}$ is called the iLLE alignment matrix and it can be constructed by $R_{\text{iLLE}} = \sum_i S_i C_i S_i^T$ where $S_i$ is a column selection matrix such that only columns $\{i_1, i_2, \ldots, i_K\}$ are chosen.

The improved LLE can be considered as a collective version of the standard LLE [14] by taking a turn for each pixel to be linearly approximated by the rest pixels in a local window. This way specifies more constraints over the matte value space. We can demonstrate that the matting based on iLLE has some significant improvement over the matting with the conventional LLE.

Xiang et al [15] also proved that the improved LLE alignment matrix can be constructed according to the local linear transformation from the image color space to the matte value space. Thus the matting algorithm based on iLLE is closely related to the Laplacian alignment matting algorithm [6]. In the construction of the Laplacian alignment matrix, the linear transformation from the color space to the matte value space is constructed based on all the color information in a local window of one pixel to predict the signal matte value corresponding to the pixel, while the iLLE alignment matrix is based $K$ local linear transformations each of which is built on color information over the rest of pixels to predict the matte value at a particular pixel in turn.

III. ALGORITHM FORMULATION

A. Formulation

In terms of machine learning terminology, the image matting is an unsupervised learning problem. As we pointed out in the introduction, one of major approaches in image matting is to use the so-called trimap or user-scribbles. The information given in a trimap or from user-scribbles is that for a group of pixels over an image, their alpha mate are known.

For this purpose, let $\Omega$ be a subset of pixels and $\Gamma_\Omega$ be the operator mapping alpha matte $\alpha$ to the given alpha matting values $\alpha_\Omega$ (trimap or user-scribbles) over $\Omega$, then the matting problem is to minimize the following objective function with respect to $\alpha$ such that $\Gamma_\Omega(\alpha) = \alpha_\Omega$
\[ \min_{\alpha} F(\alpha) = \alpha^T R \alpha. \]
where $R$ is either the LLE alignment matrix or the iLLE alignment matrix.

To get a smoother alpha matte $\alpha$, we propose to formulate two optimization problems for our image matting

1) Scribble Smoothing [6]: Instead of using hard scribble constraints, we seek for $\alpha$ by smoothing the matting value using
\[ \min_{\alpha} \alpha R \alpha^T + \lambda (\alpha - \alpha_\Omega) D_\Omega (\alpha^T - \alpha_\Omega^T) \]
(5)
where $\lambda$ is some large regularizer, $D_\Omega$ is a diagonal matrix whose diagonal elements are one for constrained pixels (in trimap or user-scribbles) and zero for all other pixels, and $\alpha_\Omega$ is the vector containing the specified alpha values for the constrained pixels and zero for all other pixels.

2) Constrained-Scribble Smoothing: In this formulation, we will constrain $\alpha$ to its soft range $0 \leq \alpha_i \leq 1$
\[ \min_{0 \leq \alpha \leq 1} \alpha R \alpha^T + \lambda (\alpha - \alpha_\Omega) D_\Omega (\alpha^T - \alpha_\Omega^T) \]
(6)
where $0 \leq \alpha \leq 1$ means the elements of the vector $\alpha$ are between 0 and 1.

B. Algorithms

It is easy to solve the optimization problem (5) for the scribble smoothing as it is an unconstrained quadratic programming problem. To reinforce the user scribble conditions $\alpha_\Omega$ over pixels $\Omega$, the algorithm in [6] takes a larger regularizer such as $\lambda = 100$. Thus the objective function can be optimized by solving a sparse linear system:
\[ (R + \lambda D_\Omega) \alpha = \lambda \alpha_\Omega. \]
(7)

However solving the optimization problem defined in (6) is much harder as it is a quadratic programming problem with a set of linear inequality constraints. One of approaches to solve a quadratic programming problem is to use an interior
point method that uses Newton-like iterations to find a solution of the Karush-Kuhn-Tucker conditions of the primal and dual problems. The computational complexity is polynomial time of the size of the matrix \( R \). Even for a moderately sized image, it is intractable for one to use a standard quadratic optimization algorithm as the number of variables is equal to the number of pixels in the whole image. Fortunately given the special form of the constraints involved in the problem, a trust-region alike algorithm as proposed in [26] can be used for the optimization problem defined in (6). The algorithm has a fast convergence rate of \( O(1/k^2) \) where \( k \) is the number of iterations. However, the trust-region alike algorithm is very expensive because the second order derivative of the objective function is needed.

However we also note that the problem is convex and smooth. In the following sections, we propose applying the optimal first-order black-box method for smooth convex optimization, i.e., Nesterov’s method [27]-[29], to achieve a convergence rate of \( O(1/k^2) \). We first construct the following model for approximating the objective function \( F(\alpha) \) in (6) at the point \( \alpha \),

\[
h_{C,\alpha}(\tilde{\alpha}) = F(\alpha) + F'(\alpha)(\tilde{\alpha} - \alpha)^T + \frac{C}{2}\|\tilde{\alpha} - \alpha\|^2, \tag{8}
\]

where \( C > 0 \) is a constant.

With model \( (8) \), Nesterov’s method is based on two sequences \( \{\alpha_k\} \) and \( \{s_k\} \) in which \( \{\alpha_k\} \) is the sequence of approximate solutions while \( \{s_k\} \) is the sequence of search points. The search point \( s_k \) is the convex linear combination of \( \alpha_{k-1} \) and \( \alpha_k \) as

\[
s_k = \alpha_k + \beta_k(\alpha_k - \alpha_{k-1})
\]

where \( \beta_k \) is a properly chosen coefficient. The approximate solution \( \alpha_{k+1} \) is computed as the minimizer of \( h_{C,s_k}(\tilde{\alpha}) \). It can be proved that

\[
\alpha_{k+1} = \arg\min_{0 \leq \alpha \leq 1} \frac{C}{2} \|\tilde{\alpha} - \left( s_k - \frac{1}{C_k} F'(s_k) \right) \|^2 \tag{9}
\]

where \( C_k \) is determined by line search according to the Armijo-Goldstein rule so that \( C_k \) should be appropriate for \( s_k \), (see [29]). \( \alpha_{k+1} \) defined by \( (9) \) is actually the projection of the vector \( s_k - \frac{1}{C_k} F'(s_k) \) over the convex set \( \{\alpha \mid 0 \leq \alpha \leq 1\} \). We can easily work out the projection given by the following formula

\[
\alpha_{k+1} = \max \left\{ 0, \min \left\{ 1, s_k - \frac{1}{C_k} F'(s_k) \right\} \right\}. \tag{10}
\]

where both max and min operate over vectors component-wisely as the same meaning in Matlab.

An efficient algorithm for (6) is summarized in Algorithm 1.

\[\text{Algorithm 1 The Efficient Nesterov’s Algorithm}\]

**Input:** \( C_0 > 0 \) and \( \alpha_0, K \)

**Output:** \( \alpha_{k+1} \)

1. Initialize \( \alpha_1 = \alpha_0, \gamma_{-1} = 0, \gamma_0 = 1 \) and \( C = C_0 \).
2. for \( k = 1 \) to \( K \) do
3. Set \( \beta_k = \frac{\gamma_{k-1}}{\gamma_k} \), \( s_k = \alpha_k + \beta_k(\alpha_k - \alpha_{k-1}) \)
4. Find the smallest \( C = C_{k-1}, 2C_{k-1}, \ldots \) such that
\[
F(\alpha_{k+1}) \leq h_{C,s_k}(\alpha_{k+1}),
\]
where \( \alpha_{k+1} \) is defined by \( (10) \).
5. Set \( C_k = C \) and \( \gamma_{k+1} = \frac{1 + \sqrt{1 + 4\gamma_k}}{2} \)
6. end for

Both \( F \) and \( B \). \( F \) and \( B \) are obtained from optimizing the following objective function

\[
\min_{F,B} \sum_i (\alpha_i F_i + (1 - \alpha_i)B_i - I_i)^2 + \langle \partial \alpha_i, (\partial F_i)^2 + (\partial B_i)^2 \rangle
\]

where \( \partial \) is the gradient operator over the image grid. For a fixed \( A \), the problem is quadratic and its minimum can be found by solving a set of linear equations.

**IV. Experiment Results**

All the algorithms are implemented using MATLAB on a small workstation machine with 32G memory. In all the calculations, we set \( \lambda = 100 \), see (7), in both algorithms.

**A. Experiment I**

In this experiment, we aim to compare the performance of our newly suggested LLE alignment matrix with the Laplacian alignment matrix introduced in [6] for image matting, under the Scribble Smoothing formulation. Similar to the closed-form solution, we solve (7) for matte values.

For this experiment, two images from the original paper describing the closed form solution for image matting [6] are used for tests. Figure 1 presents the images and their stroked images as used in the algorithm comparison.

Figure 2 presents matting results on the two images, respectively. In Figure 2, the first and third columns show the results from the closed-form solution while the second row and the fourth row show the results from the Scribble Smoothing formulation based on the LLE alignment matrix. The results given by the LLE alignment matrix look visually comparable to the results in [6].
Fig. 2. Mattes from strokes with the Closed-form solution for the Laplacian alignment matrix (the first and third columns) and the Scribble Smoothing for the LLE alignment matrix (the second and fourth columns): (a) Learned Masks (the first row). (b) Reconstructed foreground images (the second row). (c) Extracted background images (the third row).

B. Experiment II

In this experiment, we compare the performance of iLLE alignment matrix against the LLE alignment matrix based on two algorithm formulations. All the setting used in this experiment is exactly same as those used in Experiment I. Image and its scribbles used in this experiment are shown in Figure

Figure 4 presents matting results on the plant image. In Figure 4, the first and second columns show the results from the Scribble Smoothing for the LLE and iLLE alignment matrices, respectively; and the third and fourth columns show that from the Constrained-Scribble Smoothing for both LLE and iLLE alignment matrices, respectively. The results given by the iLLE alignment matrix is slightly better than those given by the LLE alignment matrix.

C. Experiment III

In this experiment, the primary goal is to assess the performance of the two formulations: Scribble Smoothing (SS) and Constrained-Scribble Smoothing (CSS) presented in this paper. We use iLLE alignment matrix as an example. For this purpose, we use benchmark test images taken from the matting website http://www.alphamatting.com. On the website, there are four types of test images. The images that we are using are (A) plastic bag (Highly Transparent); (B) net (Strongly Transparent); (C) elephant (Medium Transparent) and (D) peacock (Little Transparent but with detailed structures). The four original images and their strokes used in the experiment are displayed in Figure 5.

Fig. 3. Images and Strokes

(a) Plastic Bag (b) Net
(c) Elephant (d) Peacock

Fig. 4. Mattes from strokes with the LLE (the first and third columns) and iLLE (the second and fourth columns) alignment matrices with Scribble Smoothing (the first and second columns) and Constrained-Scribble Smoothing (the third and fourth columns) formulations: (a) Learned Masks (the first row). (b) Reconstructed foreground images (the second row). (c) Extracted background images (the third row).

Fig. 5. Images and Strokes: (a) Highly Transparent. (b) Strongly Transparent. (c) Medium Transparent. (d) Little Transparent

The results for the highly transparent image plastic bag and the strongly transparent image net are shown in Figure 6 in which the first and third columns are for the SS formulation while the second and fourth columns are for the CSS formulations. The first row shows the learnt alpha matte for each image and formulation while the second row and the third row show the extracted foreground and background images.

It is obvious that, for the highly transparent plastic bag, the learnt alpha matte from the CSS formulation is much better than the one from the SS formulation while for the strongly transparent net the results from both formulation schemes are comparable to each other.

Similarly we have shown the results for both the medium transparent elephant image and the little transparent peacock image in Figure 7.

From this group of results, we can conclude that the performance of both formulation schemes are comparable to each other, but the CSS formulation slightly outperforms
the SS formulation as can be observed from the part of background that has been absorbed into the foreground by the SS formulation, e.g., the green spot on left upper region of the peacock’s plumage in the image.

Actually the matting problem for all the images used in this experiment is very challenging. For example, the colour information of the elephant experiment is very challenging. For example, the colour information of the background and foreground in the elephant experiment is very challenging. For example, the colour information of the SS formulation as can be observed from the part of

Fig. 6. Learning Results for the Highly and Strongly Transparent Images

Fig. 7. Learning Results for the Medium and Little Transparent Images

V. CONCLUSION

This paper proposes two formulations for image matting based on the classical LLE alignment matrix and improve LLE alignment matrix. The experiments have demonstrated both formulations are comparable to each other while in many cases the CSS formulation is slightly better than the SS formulation. The experiment also demonstrated that the SS formulation based on the iLLE alignment matrix is comparable to both LEE and Laplacian alignment matrices. A similar approach [8] can be used for further edge preservation constraint. This kind of observation is intuitively encouraging for manifold learning methods to be successful in image matting.

ACKNOWLEDGMENT

The authors are partially supported by the Faculty of Business Research Compact Grants.

REFERENCES


