Drone-Based Wireless Relay using Online Tensor Update

Yao Xie*, Xiao-Yang Liu*, Linghe Kong*, Fan Wu*, Guihai Chen*, Athanasios V. Vasilakos†
*Shanghai Jiao Tong University, China
†Lulea University of Technology, Sweden
*{allenxie, yanglet, linghe.kong, fwu, gchen}@sjtu.edu.cn, †athanasios.vasilakos@ltu.se

***Abstract*—In the wireless communication, there are many cases where the transmission path is obstructed by unknown objects. With the rapid development of the drone technology in recent years, the drones are advocated to serve as mobile relays to forward data streams. However, the challenges are that data transmission may suffer severe signal attenuation due to the existence of the obstructions and it is challenging to find the best location for mobile relays due to the dynamic environment and unpredictable interference. To address the problem, this paper proposes an approach that a drone can automatically find the location with the optimal link quality. We design a novel algorithm, named Path-sampling Online Tensor Update (POTU), to estimate the link quality in the space and find the optimal location. Furthermore, the algorithm is practical to the real applications due to the simplicity of implementation. In the experiment, we construct a realistic scene and compare the performance of our algorithm with the classic and the state-of-the-art algorithms. As a result, POTU outperforms existing methods in achieving the trade-off between time cost and estimation accuracy.**

I. Introduction

The mobile relay technique is a hot research topic nowadays and has many advantages in wireless data transmission, such as its mobility for deployment. The key features of mobile relay, including its low-cost, mobility and flexibility have raised great interests in both industry and academia. In addition, due to the existence of the obstructions that lead to signal attenuation, mobile relays can be used to bypass the obstructions and enhance the quality of transmission. There are lots of use cases for mobile relay, such as group mobility, reliability, and wireless backhaul load balancing. Drones, which are also referred as unmanned aerial vehicles (UAVs), as one kind of modern mobile relays are discussed in [3] [4].

This paper addresses a realistic problem that in places where obstructions exist, the data transmission often suffers a lot of signal attenuation leading to poor transmission quality. Using mobile relays is an effective method to bypass the obstructions in the transmission path. With the rapid development of the drones and their high flexibility in real applications, the drones have become the promising option for mobile relays, especially in emergent and critical tasks, the severity of which is mentioned in [16]. For example, mobile relays can be deployed in the earthquake scene [1] to establish better communication in the disaster area. Another example is that mobile relays can provide a way to bypass the walls when high transmission quality is needed between two rooms obstructed by multiple walls. Using drones as the transmission relay is a novel way since the drone technology is in its infancy and is becoming more and more popular. Also, such real-time situation needs quick establishment of the transmission and the drones' mobility with flexibility is perfectly suitable for it.

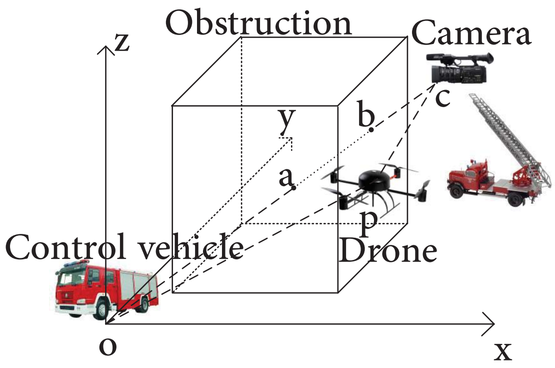


Fig. 1. Fire-fighting scenario illustration.

Specifically, in most fire-fighting scenarios, the high-quality video needs to be transmitted in real time so that the field control vehicle can make effective arrangements according to the video received from the camera in the front fire location. However, such wireless link is usually none-line-of-sight due to unknown obstructions. The video transmission path between the camera device and the field control vehicle is often obstructed by buildings, as illustrated in Fig. 1. In this situation, the path will suffer severe signal attenuation so that the control vehicle cannot receive high quality video. Setting up the mobile relay between the video transmission source and the destination is an approach to solving this problem. The drone can be used as the mobile relay in order to bypass the obstructions. It will establish a relatively longer transmission path which, however, provides much higher transmission quality. Also, due to the mobility and flexibility of the drone, it can quickly reach the relay location and the transmission between the front field camera and the field control vehicle can be established in a relatively short time.

To solve the mobile relay problem in emergent tasks, we propose an approach to finding the optimal location for mobile

1521-9097/16 $31.00 © 2016 IEEE
DOI 10.1109/ICPADS.2016.14

relays by using the drone to sample from the space and update the optimal location by an algorithm and finally arrive at the optimal location. First, we take the whole space as a 3-D tensor and treat the value of every element in the tensor as the link quality of data transmission. Such abstraction helps us better describe the environment. Then we carefully consider the data transmission model and apply the model in the tensor. Our goal is to develop a strategy for the drone to take samples and update the quality tensor with the samples and finally find the optimal location for the relay. To achieve our goal, we design an algorithm called Path-sampling Online Tensor Update (POTU). The algorithm uses an online update method to make corrections through tensor recovery iteratively with the real data sampled as inputs, resulting in rather accurate estimation of the link quality. At last, extensive simulations are conducted to compare the performance of different solutions to this problem including the classic and the state-of-the-art methods.

The contributions of this paper are listed as follows:

- To the best of our knowledge, our work is the first to focus on using tensor update method together with drones to find the optimal location for mobile relays.
- A novel algorithm, POTU, with online tensor update is designed. The algorithm is practical due to the simplicity of implementation and does not need to consider any form of distortion or interference in the dynamic environment. Rather, it utilizes the tensor update to make corrections to acquire more accuracy link qualities according to the sampled real data.
- Simulations show that POTU outperforms the classical and the state-of-the-art algorithms, providing a more balanced and stable performance.

The rest of the paper is organized as follows. In Section II, we introduce the related work in corresponding field. In Section III, we introduce the system models, describe the problem and briefly have an overview of 3-D tensors. In Section IV, we provide our approach and introduce POTU in detail. In Section V, we conduct the simulations. Our work is concluded in Section VI.

II. Related Work

Lots of works have contributed to the adoption of mobile relays [6] [7] [8] [18]. In [8], low-cost disposable mobile relays are used to reduce the total energy consumption of data-intensive wireless sensor networks (WSNs). Both centralized and distributed algorithms are proposed to find the best locations for mobile relays and minimize the total energy consumption of mobility of relays and data transmission. Using mobile relays to offer high-quality services on high-speed transportation has also been studied [7]. Key techniques, such as the group mobility, the local service support, the multi-RAT (multi-Radio Access Technology) and RAN (Radio Access Network) sharing are applied to improve the efficiency of mobile relays.

There are many other works focusing specifically on using drones, which are often referred as unmanned aerial vehicles (UAVs), as transmission relays. Drones are mentioned as a method to support the Battlefield Information System (BITS) at first, as described in [17]. It is planned to use drones as platforms for a high capacity trunk radio relay to form a drone-based battlefield broadcast system. The paper also points out that, to effectively apply the drone technique in such communication system, airborne relays using drones must maintain compatibility with the existing and planned terrestrial and space communication infrastructures.

In recent years, efforts has been also put into studying using drones as mobile relays to improve the performance of modern communication systems. In 2013, Guo and others provide an analysis on using drones to assist the cellular network performance and the experiments show that trough-to-peak throughput improvements can be acquired for users in poor coverage areas [9]. Furthermore, the paper uses stochastic geometry and multi-cell simulation results to reinforce the experimental study with large-scale network analysis.

To address the problem of establishing line-of-sight transmission path for the drones, the algorithms for relay placement are studied. In [3], two algorithms to establish the relay chain are proposed to solve the problem of using drones as transmission relay when there exists obstructions. The paper compares the two algorithms, one of which is a dual ascent algorithm and the other is a modification of the Bellman-Ford algorithm. The ground operator chooses the final placement of the drones evaluating the number of hops and the chain cost. In the simulations, both algorithms have better performances than the classic Bellman-Ford algorithm. The same authors also examine using drones as mobile relays [4]. The paper digs deeper into using drone-based surveillance where information must be transmitted to the base station in real time. The limitation of the range and the requirement of line of sight make the transmission from the distant location impossible. To solve the relay placement problem, the paper presents using graph search and a label-correcting algorithm to generate sets of Pareto-optimal chains efficiently in order to achieve trade-offs between the number of drones and the resulting quality. A dual ascent algorithm is also proposed for certain tasks and situations.

However, these solutions cannot fit in the mobile relay problem in emergent tasks since that they do not ensure the quality of real-time transmission, the bypass of obstructions and the speed of the optimal location finding. In order to find the optimal location as fast as possible for the emergency as well as ensuring the transmission quality, we need to develop a new approach.

III. Problem Formulation

We first provide our system models, then describe the problem and finally take an overview of the 3-D tensor.

A. System Models

1) Space Model: As can be seen in Fig. 1, obviously, the optimal location is somewhere between the field control vehicle and the camera device. Taking the field control car as the origin o with coordinates $(0,0,0)$ and the camera device as the point c with coordinates (n_1, n_2, n_3), we set up a 3-D space.

Accordingly, a 3-D tensor $\mathcal{Q} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ is created to represent the link quality where n_1, n_2, n_3 are the scales of the three dimensions. In this way, we split the space into $n_1 \times n_2 \times n_3$ cubes and we assume the link quality in a cube is the same. The value of each element $\mathcal{Q}(i,j,k)$ of the tensor denotes the link quality of video transmission if the drone is set in (i,j,k).

2) Link Quality Model: In order to measure the link quality for each location, we provide a model for data transmission, taking the large-scale path loss, the shadowing loss and the non-shadowing loss into consideration [11]. Let d be the distance and $P(d)$ be the large-scale path loss between the sender end and the receiver end. $Z^{(1)}$ and $Z^{(2)}$ represent the shadowing loss and the non-shadowing loss, respectively. The loss of the signal L along the transmission paths can be calculated as

$$L = P(d) + Z^{(1)} + Z^{(2)}, \tag{1}$$

where $P(d)$ is given by

$$P(d) = 10\alpha log(d) + \beta, \tag{2}$$

where α and β are constants and $\alpha \geq 2$. Furthermore, considering that attenuation parameter is not constant in the space, using the linear integral over the transmission path, for the shadowing loss $Z^{(1)}$, we have

$$Z^{(1)} = \int_{path} g(\mathbf{r}) d\mathbf{r}, \tag{3}$$

where $\mathbf{r} \in \mathbb{R}^3$ denotes a distance vector in the space and $g(\mathbf{r})$ (dB/m) denotes the attenuation parameter due to the shadowing loss on location $\mathbf{r}$. And for the non-shadowing loss $Z^{(2)}$, we assume a wide-sense stationary Gaussian process with zero mean and variance η^2.

To find a way of link quality evaluation, we should consider both the source side and the destination side. The obvious method to describe the link quality is the received transmission quality from certain points. Since the transmission quality at the destination point received from the relay point is equivalent to that at the relay point received from the destination point, we consider the receiving qualities at the relay point from both the source point and the destination point as factors in representation of link quality metrics. Let S_o, S_c denote the sending quality at the origin o and the camera point c and $R_o(i,j,k), R_c(i,j,k)$ denote the receiving quality from point o and c at point (i,j,k). According to the data transmission model above, we give the representation of the link quality $\mathcal{Q}(i,j,k)$ described as the product of the receiving qualities $R_o(i,j,k)$ at point (i,j,k) from origin point o and $R_c(i,j,k)$ at point (i,j,k) from camera point c

$$\mathcal{Q}(i,j,k) = R_o(i,j,k) \times R_c(i,j,k), \tag{4}$$

where $R_o(i,j,k)$ and $R_c(i,j,k)$ are given by

$$\begin{aligned} R_o(i,j,k) &= S_o - L_o, \\ R_c(i,j,k) &= S_c - L_c, \end{aligned} \tag{5}$$

where S_o, S_c are the sending quality at o and s and L_o, L_c are the signal loss along the path from point o and c, respectively. Here we want to achieve better total performance of the relay transmission quality and it can be easily proven in mathematics that the product of the two value is the best way to evaluate it.

Let H denote the set of locations covered by the transmission path and the shadowing loss of the signal can be given by

$$Z^{(1)} = \Sigma_{(x,y,z) \in H} g(x,y,z) r(x,y,z), \tag{6}$$

where $g(x,y,z)$ is the corresponding attenuation parameter of the cube (x,y,z) and $r(x,y,z)$ is the path length in the cube (x,y,z).

B. Problem Description

The drone aims to find the optimal location with the best link quality. We need to develop a strategy of path choosing for the drone and a method to update the quality tensor $\mathcal{Q}$ in an online manner.

In this problem, the drone flies along the path according to the strategy in the space with the initialization of the quality tensor $\mathcal{Q}$ using the data transmission model. Thus, the problem is described as follows:

In the space set up based on the destination point $o(0,0,0)$ and the source point $c(n_1, n_2, n_3)$, given the start point $s(s_x, s_y, s_z)$ and the initial quality tensor $\mathcal{Q}$, we design a strategy for the drone to find the optimal point $p(p_x, p_y, p_z)$ so that the link quality $\mathcal{Q}(p_x, p_y, p_z)$ is maximized.

For the reason that such strategy is applied in the certain scenes for the emergency situations such as fire-fighting, the energy consumption for the drone is not considered as the battery for the drone is assumed to be enough for the completion of most fire-fighting scenarios. Due to the current development of drone technology, the stability of the drone is also not considered. Other factors such as frequency spectrum are excluded since the specific application scenarios are short-ranged and these factors are much less significant in this problem.

C. Overview of 3-D Tensors

To solve the problem, we first introduce some concepts in the linear algebra and present the general statement of the tensor completion problem. For a 3-D tensor $\mathcal{X}$, $\mathcal{X}^{(i)}$ denotes the ith frontal slice of $\mathcal{X}$. We use $\mathcal{X}(:,:,k)$, $\mathcal{X}(:,j,:)$ and $\mathcal{X}(i,:,:)$ to denote the kth frontal slice, jth lateral slice and ith horizontal slice, respectively. $\hat{\mathcal{X}}$ denotes the 3-D tensor obtained by taking the Fourier transform on $\mathcal{X}$ along the third dimension.

1) Linear operation: T-product is used to define the multiplication for 3-D tensor. For a 3-D tensor $\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, we can view it as an $n_1 \times n_2$ matrix with each element as a tube which can be seen as a vector into the tensor. Each tube of the 3-D tensor can be described as $\mathcal{X}(i,j,:)$. Then we give the definition of t-product.

Definition III.1. ***t-product.*** The t-product $\mathcal{C}$ of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} \in \mathbb{R}^{n_2 \times n_4 \times n_3}$ is a tensor of size $n_1 \times n_4 \times n_3$ where the $(i,j)_{th}$ tube denoted by $\mathcal{C}(i,j,:)$ for $i = 1,2,...,n_1$ and $j = 1,2,...,n_4$ of the tensor $\mathcal{C}$ is given by $\Sigma_{k=1}^{n_2} \mathcal{A}(i,k,:) * \mathcal{B}(k,j,:)$.

The t-product of $\mathcal{A}$ and $\mathcal{B}$ can be calculated by performing the Fast Fourier transformation (FFT) along the tubes of $\mathcal{A}$ and $\mathcal{B}$ to obtain $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$, then multiplying each pair of the frontal slices of $\hat{\mathcal{A}}$ and $\hat{\mathcal{B}}$ to obtain $\hat{\mathcal{C}}$, and finally performing the inverse FFT along the third dimension to obtain the result. The details are shown in [10] [12] [13].

2) t-SVD: Using multiplication for 3-D tensors, we can compute a tensor-Singular Value Decomposition (t-SVD). First, we give the following definitions.

Definition III.2. ***Tensor Transpose.*** For a tensor $\mathcal{X}$ with size $n_1 \times n_2 \times n_3$, $\mathcal{X}^\top$ is obtained by transposing each of the frontal slices and reversing the order of transposed frontal slices 2 through n_3.

Definition III.3. ***Identity Tensor.*** The identity tensor $\mathcal{J} \in \mathbb{R}^{n \times n \times n_3}$ is tensor whose first frontal slice is the $n \times n$ identity matrix and all other frontal slices are zero.

Definition III.4. ***Tensor Tubal-rank.*** The tensor tubal-rank of a 3-D tensor is the number of non-zero tubes of $\mathcal{S}$ in the t-SVD factorization.

For a tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, the t-SVD of $\mathcal{M}$ is given by

$$\mathcal{M} = \mathcal{U} * \mathcal{S} * \mathcal{V}^\top, \tag{7}$$

where the tensors $\mathcal{U}$ and $\mathcal{V}$ have the feature that $\mathcal{U}^\top * \mathcal{U} = \mathcal{J}$, $\mathcal{V}^\top * \mathcal{V} = \mathcal{J}$.

Using the t-SVD, we can extract notions of complexity of the data in the matrix in terms of "rank". The notion of multi-rank was proposed in [12] using the Fourier Domain representation of t-SVD as the vector of ranks of the slices $\hat{\mathcal{X}}(:,:,k), k = 1,2,...,n_3$. The l_1 norm of the multi-rank can then be a way to measure the complexity of the data in the matrix.

3) Tensor Completion: Compressive sensing is usually a method for tensor completion [14] [15]. Consider the problem to predict the missing data of a tensor with some of its data being sampled. Suppose there is an unknown tensor $\mathcal{M} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ which is assumed to have a low tubal-rank, which indicates that in continuous space, the elements have high relativity. A subset of entries $\{\mathcal{M}_{i,j,k} : (i,j,k) \in \Omega\}$ is sampled where Ω is an indicator tensor of size $n_1 \times n_2 \times n_3$ meaning which entries of the tensor are being obeserved. The goal is to recover the entire tensor $\mathcal{M}$ from sampled data. To address the problem of tensor completion, the following minimization problem should be solved

$$\min ||\mathcal{X}||_{TNN}, \text{ subject to } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{M}), \tag{8}$$

where $||\mathcal{X}||_{TNN}$ denotes the tensor-nuclear-norm (TNN) and is defined as the sum of the singular values of all the frontal slices of $\hat{\mathcal{X}}$ [19] and P_Ω is the orthogonal projector indicating the sampled location. Therefore, the component (i,j,k) of $P_\Omega(\mathcal{X})$ equals to the component (i,j,k) of $\mathcal{M}$ if $(i,j,k) \in \Omega$ and zero otherwise.

IV. Path-sampling Online Tensor Update

Based on the 3-D tensor, we propose the Path-sampling Online Tensor Update (POTU) algorithm to solve the problem. In this section, we first take an overview on our approach and then show the details of our algorithm.

A. Overview

1) Initial Setup: We assume that point o, c, s are known initially. These points are obtained by the drone using the camera to observe the location of the field control vehicle and the camera device. Then we set up the corresponding 3-D space taking the field control vehicle as the origin $o(0,0,0)$ and the location of the camera as $c(n_1,n_2,n_3)$. Accordingly, the initial point $s(s_x,s_y,s_z)$ where the drone starts is also set, usually, to a point on the ground.

Second, the initial values for the 3-D quality tensor $\mathcal{Q}$ are set. Considering that the shadowing loss and the non-shadowing loss of the signal are relatively less compared to the large-scale path loss between the sender end and the receiver end, we initialize the quality tensor with values only taking the large-scale path loss into account leading to $L = P(d)$. Note that any form of the distortion on the transmission link quality resulted from the dynamic environment is corrected in the tensor update using the sample real data, so that the initialization does not consider the distortion or the attenuation. Thus, by (4) and (5), for each element in the quality tensor, the value is initialized as

$$\begin{aligned} \mathcal{Q}(i,j,k) &= R_o \times R_c \\ &= (S_o - L_o) \times (S_c - L_c) \\ &= (S_o - P(d_o)) \times (S_c - P(d_c)), \end{aligned} \tag{9}$$

where d_o and d_c represent the distances between the drone and the vehicle, the camera respectively and the large-scale path loss $P(d_o)$, $P(d_c)$ are given by (2). The distances d_o and d_c are given by $d_o = \sqrt{i^2 + j^2 + k^2}, d_c = \sqrt{(i-n_1)^2 + (j-n_2)^2 + (k-n_3)^2}$.

2) Optimal Location Finding Procedure: In the optimal location finding procedure, samples are taken for updating the link quality tensor. To acquire the link quality of each point, signals are sent by both the field control vehicle and the camera device to the drone. The data contained in the signals includes the information about the sender, such as the original signal strength S at the sender end and the transmission start time. By measuring the signal strength R at the receiver end, we can get the signal loss $L = S - R$ along the path. After getting the receiving quality R_o from the field control vehicle point o and R_c from the camera device point c, the link quality value for the current location is calculated by (5) and the drone moves to the next location.

We develop a strategy for the drone to take samples for updating the quality tensor and finally find the optimal location, ensuring an accurate estimation of the tensor. Let the set D

be the data sample set and d_u denote the data sampled at the location $u(u_x, u_y, u_z)$. By the initialization of the 3-D tensor, regardless of the shadowing loss and the non-shadowing loss whose influences will be corrected by tensor update using real sample, we have already had general estimation of the link quality tensor, which lacks accuracy for certain. However, it is good enough for an initialization. Thus, the drone follows the below steps:

- Step 1: Fly to the current optimal location and take samples.
- Step 2: Update the link quality tensor $\mathcal{Q}$ with D together with $\mathcal{Q}$.
- Step 3: If the current location is the optimal location, end the procedure. If not, repeat from Step 1.

After every round of updating the link quality tensor, if the drone doesn't arrive at the optimal location, it continues to find the optimal location and takes samples along the way. The exact data of each location is used to update the link quality tensor based on the value achieved before. Specifically, the values taken by the drone and the values previously achieved together are the input of the tensor update procedure to make corrections on the achieved quality tensor. In this way, gradually, we achieve a more accurate link quality tensor for optimal location finding.

B. Algorithm

To solve the problem, we design an algorithm named Path-sampling Online Tensor Update (POTU) to find the optimal location. The detail of the algorithm is shown in Algorithm 1. In general, when the current location is not the optimal location, POTU samples the data along the path to the optimal location and updates the link quality tensor using TensorUpdate method which is introduced later. Then POTU finds the current optimal location and determines whether to terminate or to continue the next iteration.

Algorithm 1 Path-sampling Online Tensor Update

```
Input: $o, c, s, \mathcal{Q}$
Output: $p$
1:  $D \leftarrow \emptyset$
2:  Main Procedure
3:  while $s \neq p$ do
4:      for every point $u$ along the path from $s$ to $p$ do
5:          $D \leftarrow D \cup d_u$
6:      end for
7:      $Q \leftarrow$ TensorUpdate$(Q, D)$
8:      $s \leftarrow p$
9:      $p \leftarrow \operatorname*{argmax}_v \{Q(v_x, v_y, v_z)\}$
10: end while
```

Initially, the input points, the starting point s, the origin point o and the camera point c are obtained by the camera carried by the drone itself and set accordingly. The 3-D space is set up according to o and c. As for the initialization of the quality tensor $\mathcal{Q}$, the area where the transmission path is shadowed by the obstructions like buildings is considered to have relatively high signal loss so that the locations in such area cannot be the optimal relay location and are not considered in our problem. Note that the set D which records the data that has been sampled till the current time is initialized to be empty as presented in step 1 in the algorithm.

In the main procedure, POTU takes samples along the path and updates the link quality tensor $\mathcal{Q}$. As long as that the current location is not the optimal location, the drone continues to fly to the optimal location and samples data for the next round of tensor update. Such procedure is implemented by a while loop from step 3 to step 10 in the algorithm. Note that after each process of flying to the current optimal location, the number of data observed increases.

By taking the data sampled by the drone and the previously achieved values as inputs, we define tensor update procedure TensorUpdate$(\mathcal{Q}, D)$ in step 7 in the algorithm as a tensor recovery problem, which actually makes corrections on the previously achieved value by using the observed value. Considering the tensor recovery problem, for an unknown 3-D link quality tensor $\mathcal{Q}$, the goal is to recover the whole tensor $\mathcal{Q}$ from the selected entries by solving the problem

$$\min \; ||\mathcal{X}||_{TNN}, \text{ subject to } P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{Q}), \tag{10}$$

which is in the same form as (8). Here, all the entries in the tensor are selected and the exact sampled data is used for entries at which locations have been sampled and the previously achieved value is used for other entries.

Let $\mathcal{Y}$ be the sampled data, we have $\mathcal{Y} = P_\Omega(\mathcal{Q})$. Define $\mathcal{F}_3$ and $\mathcal{F}_3^{-1}$ to be the Fourier and inverse Fourier transform along the third dimension and $\mathcal{G} = \mathcal{F}_3 P_\Omega \mathcal{F}_3^{-1}$ to be an operator. Thus, we have $\hat{\mathcal{Y}} = \mathcal{G}(\hat{\mathcal{M}})$ where $\hat{\mathcal{Y}}$ and $\hat{\mathcal{M}}$ are the Fourier transforms of $\mathcal{Y}$ and $\mathcal{M}$ along the third dimension. Under such construction, the problem (10) can be then represented as follows:

$$\min \; ||\text{blkdiag}(\hat{\mathcal{X}})||_*, \text{ subject to } \hat{\mathcal{Y}} = \mathcal{G}(\hat{\mathcal{X}}), \tag{11}$$

where blkdiag$(\hat{\mathcal{X}})$ is a block diagonal matrix whose diagonal blocks are given by $\hat{\mathcal{X}}^{(i)}$. Note that $||\mathcal{X}||_{TNN} = ||\text{blkdiag}(\hat{\mathcal{X}})||_*$. The optimization problem can be rewritten as follows:

$$\begin{aligned} \min \; & ||\text{blkdiag}(\hat{\mathcal{Z}})||_* + \mathcal{L}_{\hat{\mathcal{Y}}=\mathcal{G}(\hat{\mathcal{X}})}, \\ & \text{subject to } \hat{\mathcal{X}} - \hat{\mathcal{Z}} = 0, \end{aligned} \tag{12}$$

where $\mathcal{L}$ denotes the indicator function. By applying the framework of Alternating Direction Method of Multipliers (ADMM) [2], we can have the following recursion

$$\begin{aligned} \hat{\mathcal{X}}^{k+1} &= \operatorname*{argmin}_{\hat{\mathcal{X}}} \{\mathcal{L}_{\mathcal{Y}=\mathcal{G}(\hat{\mathcal{X}})} + \langle \hat{\mathcal{T}}^k, \hat{\mathcal{X}} \rangle + \frac{1}{2}||\hat{\mathcal{X}} - \hat{\mathcal{Z}}^k||_2^2\} \\ &= \operatorname*{argmin}_{\hat{\mathcal{X}}:\mathcal{Y}=\mathcal{G}(\hat{\mathcal{X}})} \{||\hat{\mathcal{X}} - (\hat{\mathcal{Z}}^k - \hat{\mathcal{T}}^k)||_F\}, \end{aligned} \tag{13}$$

$$\begin{aligned} \hat{\mathcal{Z}}^{k+1} = \operatorname*{argmin}_{\mathcal{Z}} \{ & \frac{1}{\rho}||\text{blkdiag}(\hat{\mathcal{Z}})||_* + \\ & \frac{1}{2}||\hat{\mathcal{Z}} - (\hat{\mathcal{X}}^{k+1} + \hat{\mathcal{T}}^k)||_F\}, \end{aligned} \tag{14}$$

$$\hat{\mathcal{T}}^{k+1} = \hat{\mathcal{T}}^{k} + (\hat{\mathcal{X}}^{k+1} - \hat{\mathcal{Z}}^{k+1}). \tag{15}$$

The solution to (14) is given by the singular value thresholding [5]. By examining the format of (14), it can be split into n_3 minimization sub-problems. Let $\hat{\mathcal{Z}}^{k+1,(i)}$, $\hat{\mathcal{X}}^{k+1,(i)}$, $\hat{\mathcal{T}}^{k,(i)}$ denote the ith frontal slice of $\hat{\mathcal{Z}}^{k+1}$, $\hat{\mathcal{X}}^{k+1}$ and $\hat{\mathcal{T}}^{k}$. Then (14) can be split into:

$$\begin{aligned} \hat{\mathcal{Z}}^{k+1,(i)} = \underset{W}{\operatorname{argmin}}\{&\frac{1}{\rho}||W||_* + \frac{1}{2}||W \\ &- (\hat{\mathcal{X}}^{k+1,(i)} + \hat{\mathcal{T}}^{k,(i)})||_F\}, \end{aligned} \tag{16}$$

for $i = 1, 2, ..., n_3$. In this way, we can calculate every ith slice of $\hat{\mathcal{Z}}^{k+1}$. By solving (16), we can update the link quality tensor presented in Step 7 in the algorithm.

After every round of tensor update procedure, the drone sets the current point as the start point s and by traversing the achieved quality tensor, the drone finds the location with the best link quality and sets the new optimal location p as presented in step 8 and step 9. Afterwards, the drone examines whether the optimal location p is the same as the current start point s. If it is, then the optimal location p is found and the while loop ends. If not, the drone flies to the optimal location and makes the next round of correction.

Furthermore, it can be predicted that the whole loop is certain to come to an end. Since the initialization has found a relatively good location for video transmission relay, the drone actually takes samples in the certain small area around the first found optimal location. Therefore, POTU terminates in limited time.

V. Simulation

In this section, we evaluate the performance of POTU by comparing POTU with the classic and the state-of-the-art algorithms.

A. Environment Setting

In the simulations, we create a 3-D space scene which simulates the real life scene. Specifically, in the 3-D space based on the point $o(0, 0, 0)$ and $c(n_1, n_2, n_3)$, we put a cuboid representing the building in the environment marked by $b_1, b_2, ..., b_8$ based on the value of n_1, n_2, n_3. The settings of these points are listed in the Table I. The start point s is set to $(n_1, n_2, 0)$, one point on the ground.

Since the signal attenuation through the obstruction is severe, we assume the optimal location for mobile relay does not lie in the location where transmission path is obstructed by the building. Thus, these points are ruled out when finding the optimal location for best link quality. To simulate the real life scene, in addition to considering the large-scale path loss, we add a random value with normal distribution to obtain the experiment data. Additionally, the values in the tensor have all been normalized by dividing the value by the maximum value in the tensor.

In the simulation, we first consider the base methods to achieve the optimal location. We traverse the whole real quality tensor to find the real optimal location with the best link quality. Afterwards, we compare the performance of POTU with that of the classic algorithm, K-Nearest Neighbor (KNN), and that of the state-of-the-art algorithm Tensor Recovery (TR). In KNN, during the flying process to the optimal location, the drone updates the values of the 6 nearest neighbors of the current location and finds the new optimal location until reaching the final optimal location. While in TR, the link quality tensor is recovered once with the sampled data then the optimal location is determined.

TABLE I
Obstruction points setting in the simulation.

b_1	$(0, \frac{1}{4}n_2, 0)$
b_2	$(\frac{3}{4}n_1, \frac{1}{4}n_2, 0)$
b_3	$(\frac{3}{4}n_1, n_2, 0)$
b_4	$(0, n_2, 0)$
b_5	$(0, \frac{1}{4}n_2, n_3)$
b_6	$(\frac{3}{4}n_1, \frac{1}{4}n_2, n_3)$
b_7	$(\frac{3}{4}n_1, n_2, n_3)$
b_8	$(0, n_2, n_3)$

To evaluate the final optimal location quality, we conducted T trials and in each trial, N simulations were done and we measure the performance E of each trial by

$$E = \Sigma_{i=1}^{N}(Q_i - Qr)^2, \tag{17}$$

where Qr is the link quality of the real optimal location. The metric E evaluates the difference between the achieved results and the real optimal location. Better algorithms have smaller value since the achieved results are closer to the link quality of the real optimal location. We calculate the average of the trial performances E to compare the general performance of the three algorithms.

In addition, to compare the stability of the three algorithms, the standard error is calculated by

$$Error_s = \sqrt{\frac{\Sigma_{i=1}^{N}(Q_i - \bar{Q})^2}{N-1}}, \tag{18}$$

where $\bar{Q} = \frac{\Sigma_{i=1}^{N}Q_i}{N}$. The metric $Error_s$ evaluates the standard error of the results of different simulations. The smaller the standard error is, the more stable results the algorithm produces.

In order to measure the efficiency, the time costs of the three algorithms are also compared. We evaluate the time cost for each trial by the average time cost of each simulation and the formulation is given by

$$\bar{t} = \frac{\Sigma_{i=1}^{N}t_i}{N}. \tag{19}$$

Experiments are conducted with different data sizes to acquire more results about the performance.

The procedures of simulation are as follows:

- First, initialize with the parameters of the original data. Setup the 3-D space, mark the point o, c, s and calculate the initial quality tensor $\mathcal{Q}$.
- Second, simulate the drone's flying process. Then use the algorithms POTU, KNN and TR to update the link quality tensor and record the link quality of the final location for relay.
- Finally, repeat the simulations several times with different parameters and compare the performances among the algorithms.

B. Simulation Results

In Fig. 2, we plot the performance comparisons among the three algorithms on multiple trials and with different tensor sizes.

To evaluate the quality performance, it can be seen from (a), (b) and (c) that, in general, POTU outperforms KNN and TR. According to our measurement of the link quality which is given by (17), we know that the less the performance E is, the better the link quality is acquired. Specifically, in (a), POTU achieves the performance value 0.412×10^{-3}, KNN achieves 1.289×10^{-3} and TR achieves 0.703×10^{-3}. Comparing the results achieved by the three algorithms, we find that POTU is about 3 times better than KNN and 2 times better than TR. In (c), where the performance of TR is close to POTU whose values are 0.586×10^{-3} and 0.430×10^{-3}, respectively, POTU is about 5 times better than KNN whose value is 1.935×10^{-3}. From (b), POTU also has a smaller value of E indicating better performance than TR and KNN. In total, it can be seen from the first three figures that performance of POTU is better than KNN and TR.

Consider the stability of the algorithms, POTU has higher stability. Specifically, in (a), the values of $Error_s$ for each algorithm are 0.189×10^{-3} for POTU, 0.735×10^{-3} for TR and 0.822×10^{-3} for KNN. POTU produces less fluctuation in achieved quality performances than TR and KNN. While in (b), the corresponding values of $Error_s$ are 0.238×10^{-3} for POTU, 0.441×10^{-3} for TR and 0.635×10^{-3} for KNN, from which we can easily conclude that the results produced by POTU have relatively smaller standard error and thus POTU is more stable. In (c), similarly, POTU gives a more stable results than TR and KNN which have relatively more fluctuation in the produced results. With different data sizes, POTU also provides a more stable performance than KNN and TR. When measuring the stability of the algorithms, POTU outperforms KNN and TR.

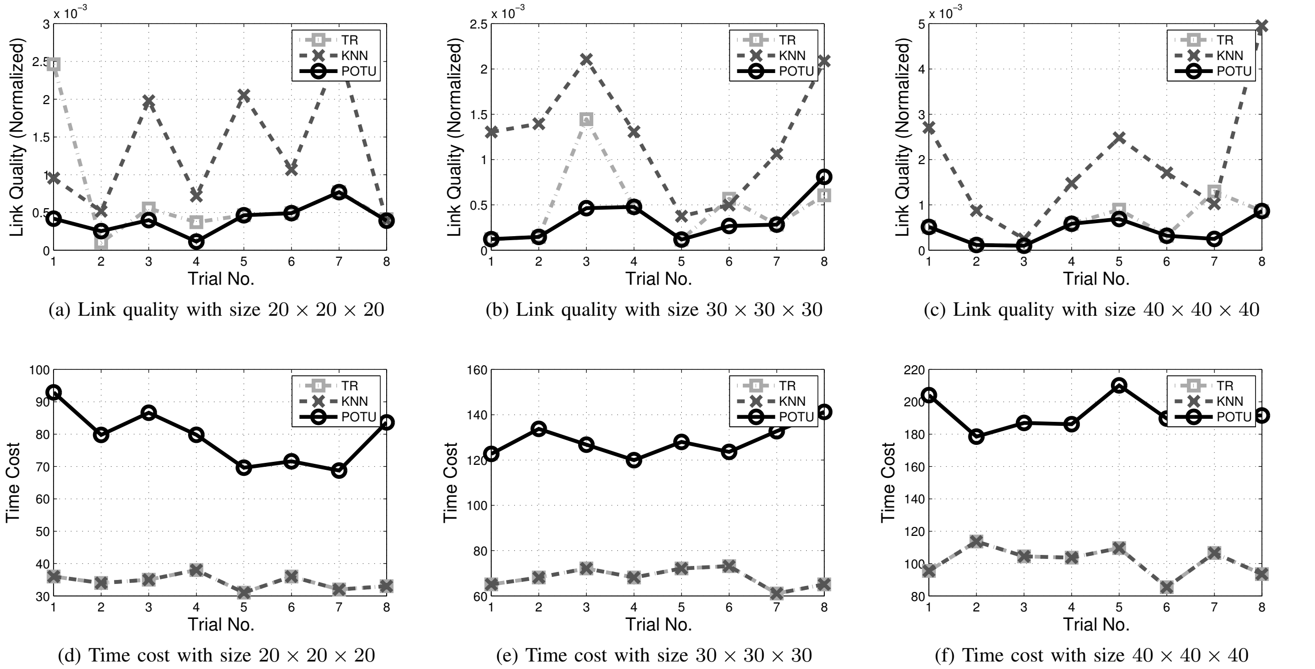
Evaluating the time costs of the three algorithms, POTU costs more than KNN and TR. From (a), we can calculate the time costs. For POTU, the average time cost is 79 while for KNN and TR, the average time costs are around 35. From (b) and (c), it can be seen that the time cost of POTU is about 2 times the time costs of KNN and TR. In general, POTU has relatively more time cost than KNN and TR. It cost more time for POTU to reach the final optimal location for transmission delay, however, the transmission can be established before reaching the optimal location. That is, shortly after taking off from the starting point, say when reaching the relay location of the first iteration of the algorithm, the drone starts to transmit data from the source to the destination and continues to further iterations at the same time. Under such circumstances, the actual transmission begins much earlier than reaching the final optimal location which makes up the disadvantage of POTU in time cost.

In conclusion, POTU outperforms KNN and TR in acquiring better quality and achieving higher stability, but with more time cost. POTU produces better transmission relay location than TR and much better results than KNN. As for stability of the results, POTU has great advantage over KNN and also is more stable than TR. When it comes to time cost, since extra time consumed by POTU is used for the link quality tensor update to achieve better location and the transmission can be established before reaching the final optimal location while using POTU, the significance of the time cost is less than the quality performance or the stability. All in all, POTU outperforms KNN and TR.

VI. Conclusion

In this paper, we focus on using mobile relays to enhance the signal quality in data transmission. Considering the existence of obstructions along the data transmission path, we propose to use the drones to forward the data stream in order to bypass the obstruction. Based on such method, we design the POTU algorithm for the drone to sample data from the environment and update the link quality tensor to find the optimal location for mobile relay. Simulations on the POTU, KNN and TR show that the POTU algorithm finds better location for transmission relay and produces more stable results but with relatively more time cost. As for the time cost, however, the drone can start transmission shortly after taking off from the start point and continue to find the optimal location at the same time. In conclusion, POTU has better performance in achieving the trade-off between time cost and optimal location finding.

The future works on such topic are as follows. First, more effort can be put into setting up a more realistic initialization for the link quality tensor so that during the updating process, it will be easier to find the optimal location. For example, frequency spectrum can be considered when establishing the transmission and proper estimation of attenuation can be considered when calculating the initial link quality tensor in order to speed up the process. Second, more other factors can be considered into the update algorithm to improve the update accuracy and the performance stability. For instance, certain conditions of the environment affecting the parameters of the space can be considered such as humidity and temperature. Also, dynamic self-adjusted parameters can be used within the algorithm to acquire more accurate estimation of the link quality tensor.



(a) Link quality with size 20 × 20 × 20

(b) Link quality with size 30 × 30 × 30

(c) Link quality with size 40 × 40 × 40

(d) Time cost with size 20 × 20 × 20

(e) Time cost with size 30 × 30 × 30

(f) Time cost with size 40 × 40 × 40

Fig. 2. Simulation results for POTU, KNN and TR.

Acknowledgment

This research was supported in part by NSFC grant 61672349, 61303202, and China Postdoctoral Science Foundation grant 2014M560334, 2015T80433. The authors would also like to thank Mr. Yifeng Cao for his valuable comments.

References

[1] K. Anazawa, P. Li, T. Miyazaki, and S. Guo. Trajectory and data planning for mobile relay to enable efficient internet access after disasters. In *2015 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2015.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[3] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, and P.-M. Olsson. Positioning unmanned aerial vehicles as communication relays for surveillance tasks. In *Robotics: Science and Systems*, 2009.

[4] O. Burdakov, P. Doherty, K. Holmberg, J. Kvarnström, and P.-M. Olsson. Relay positioning for unmanned aerial vehicle surveillance. *The international journal of robotics research*, 2010.

[5] J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[6] B.-J. Chang, Y.-H. Liang, and S.-S. Su. Analyses of qos-based relay deployment in 4g lte-a wireless mobile relay networks. In *2015 21st Asia-Pacific Conference on Communications (APCC)*, pages 62–67. IEEE, 2015.

[7] L. Chen, Y. Huang, F. Xie, Y. Gao, L. Chu, H. He, Y. Li, F. Liang, and Y. Yuan. Mobile relay in lte-advanced systems. *Communications Magazine, IEEE*, 51(11):144–151, 2013.

[8] F. El-Moukaddem, E. Torng, G. Xing, and G. Xing. Mobile relay configuration in data-intensive wireless sensor networks. *Mobile Computing, IEEE Transactions on*, 12(2):261–273, 2013.

[9] W. Guo, C. Devine, and S. Wang. Performance analysis of micro unmanned airborne communication relays for cellular networks. In *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2014 9th International Symposium on*, pages 658–663. IEEE, 2014.

[10] N. Hao, M. E. Kilmer, K. Braman, and R. C. Hoover. Facial recognition using tensor-tensor decompositions. *SIAM Journal on Imaging Sciences*, 6(1):437–463, 2013.

[11] H. Hashemi. The indoor radio propagation channel. *Proceedings of the IEEE*, 81(7):943–968, 1993.

[12] M. E. Kilmer, K. Braman, N. Hao, and R. C. Hoover. Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 34(1):148–172, 2013.

[13] M. E. Kilmer and C. D. Martin. Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641–658, 2011.

[14] L. Kong, L. He, X.-Y. Liu, Y. Gu, M.-Y. Wu, and X. Liu. Privacy-preserving compressive sensing for crowdsensing based trajectory recovery. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pages 31–40. IEEE, 2015.

[15] L. Kong, M. Xia, X.-Y. Liu, G. Chen, Y. Gu, M.-Y. Wu, and X. Liu. Data loss and reconstruction in wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 25(11):2818–2828, 2014.

[16] C. H. Lee, I. Orikumhi, C. Y. Leow, M. A. B. Malek, and T. A. Rahman. Implementation of relay-based emergency communication system on software defined radio. In *Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on*, pages 787–791. IEEE, 2015.

[17] M. F. Pinkney, D. Hampel, and S. DiPierro. Unmanned aerial vehicle (uav) communications relay. In *Military Communications Conference, 1996. MILCOM'96, Conference Proceedings, IEEE*, volume 1, pages 47–51. IEEE, 1996.

[18] Y. A. S. Yudo, N. Shigei, and H. Miyajima. Effective initial route construction for mobile relay on wireless sensor network. *Artificial Life and Robotics*, 20(1):49–55, 2015.

[19] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer. Novel methods for multilinear data completion and de-noising based on tensor-svd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3842–3849, 2014.