ABSTRACT: We describe in this paper, at a high level, a new approach to archive or store movies (or other media content) in the base content of DNA and to retrieve them without error. This approach provides compact storage for thousands of years without fear of obsolescence since DNA is a universal information storage mechanism in biological organisms.

1. THE BASIC IDEA

You might remember DNA from your biology class in high school or college. DNA stands for Deoxyribonucleic Acid and it is a molecule that encodes the genetic instructions used in living organisms. DNA consists of two long chains of nucleotide structures which are the chromosomes. DNA is most famous for its double helix structure, which was first revealed in a series of papers in 1953 [13]. The double helix has since become an iconic image and instantly recognizable visual representation of genetics and biology, or even science in general—and not surprisingly it often shows up in movies: on computer screens and holographic displays, and in science labs with genetic testing and experiments. Think “radiactive spiders” or “lost dinosaurs”, for example, the stuff of science labs with genetic testing and experiments. Think “movie researchers” to bring dinosaurs back to life in “Jurassic Park”.

Why would you want to do that? And number two: How would we store movies in DNA?

Two questions might be popping in your mind. Number one: Why would you want to do that? And number two: How would we store movies in DNA?

Storing movies in DNA

The process to store and retrieve movies on DNA is, conceptually at least, quite simple and it proceeds through the following steps:

1. Digitize: Starting from the original content (movie, show, or other), digitize the content to obtain a string of 0’s and 1’s.
2. Encode: Recall that DNA molecules consist of two strands coiled around each other to form the double helix mentioned earlier. The two strands are in turn composed of 4 types of nucleotides referred to as A, C, T, and G. This second step then is to take the sequence of 0’s and 1’s obtained at the end of step 1 and convert it into a sequences of nucleotides. One simple way to do it (although not recommended in practice) would be to map bits one-to-one with nucleotides, for example 0 randomly to A or C, and 1 randomly to T or G. The output of step 2 then is a long sequence of A, C, T, and G’s.
3. Synthesize: This step takes the string of A, C, T, G obtained at the end of step 2 and creates or synthesizes artificial DNA (meaning non-biological DNA) with the same sequence of nucleotides, using commercially available synthesis machines.
4. Archive: The DNA so obtained at the end of step 3, for however long is needed.
5. Sequence: Read the stored DNA strands using commercially available DNA sequencing machines and thus getting back a sequence of A, C, T, and G’s.
6. Decode: Using the inverse coding technique from that used in step 2, convert the sequence obtained at the end of step 5 into a sequence of 0’s and 1’s.
7. Read: The sequence of 0’s and 1’s and play the movie or, more specifically, the synthesizable amount of oligos (“Oligos”) and the amount of base-pairs that can be sequenced for one dollar (“seq $bp”). A linear curve in Figure 1 indicates an exponential increase in cost with an increase in the number of sequenced base-pairs, as shown in Figure 3 below (from [9]).

In parallel with increased capabilities, technological advances have led to a massive decrease in the cost to synthesize or sequence DNA. This is visible in Figure 1 in the curve labelled “seq $bp” and also illustrated in Figure 2, which shows the really amazing decrease in sequencing costs, even compared to a Moore-law-type decrease.

2. TRENDS IN BIO-TECHNOLOGY

Biotechnology is a recent story, which emerged around the time of the arrival of the transistor. The double-helix model of DNA was revealed, along with experimental supporting evidence, in a series of five articles in Nature [3]. A few years later in 1977 Sauvage et al. described a sequencing method to map the DNA of a complete bacterium genome. The Human Genome Project was launched in 1984 and completed in 2001; two years earlier than planned at a cost estimated to be around $3 billion [4]. Human genome sequencing today takes a few days and costs less than $10,000, with the price steadily dropping down.

The exponential increases in transistor and integrated circuit capabilities have been summarized using the celebrated Moore’s law, which observes that the number of transistors in an integrated circuit doubles approximately every two years. The speed of computer processing has far better than doubled every two years since 1980, in other words “butter-to-Moore’s law has gone berserk”. This is illustrated in Figure 1 (adapted from Reference [7]) which shows on the log-scaled y-axis the relative growth in capabilities over time for both DNA synthesizing (writing) and sequencing (reading), with the synthesizable content of DNA (writing) far surpassing the synthesizable amount of oligos (“Oligos”) and the amount of base-pairs that can be sequenced for one dollar (“seq $bp”).
Much of the focus in biotechnology until recently has been on medical applications and use cases. Church et al. in 2012 published a successful experiment to store a 0.5 gigabyte book in DNA [3], [4], with a somewhat similarly-sized scale-up by Jou谐 at al. in 2013 documented in [3]. Grass et al. in 2015 showed that DNA would be a reliable storage media for long-term archiving of digital data [9]. Altogether, these early results along with the subsequent described earlier of rapidly decreasing costs for DNA sequencing suggest that storing large, valuable data such as movie archives in DNA will become feasible in the future. Some key challenges remain, as mentioned earlier, in particular the development of coding mechanisms to handle and correct errors during DNA synthesis and sequencing, and the development of efficient methods to synthesize large amounts of DNA, but efforts underway (such as that described in this paper) indicate that indeed, DNA movie archiving will very likely happen in the next several years.

3. STORING MOVIES IN DNA

At each position in an oligo, 4 different nucleotides (A, C, T, G) can be placed—in other words each position in an oligo can represent 2 bits of data. Since arbitrary oligos can be synthesized and sequenced, arbitrary sequences of digital data can be stored in DNA molecules. However, current synthesis technology can only store indices, which together means a payload per oligo of 46 bits. We evaluated the performance of our scheme using a detailed simulation model of the end-to-end DNA archival process. Assuming a book of oligos of N oligos and a wide range of error rates, as well as a Gaussian coverage (meaning significant lack of coverage on the tails), we found that only 6 to 9 randomly taken samples of 8 oligos have to be iteratively sequenced until the stored data can be recovered completely and error-free. Note that the coupon collector’s problem is a good analogy for this.

2. CODING SCHEME TO HANDLE SYNTHESIS AND SEQUENCING ERRORS

Our coding scheme to address the issues mentioned in Section 3. We describe one of them in more detail here.

The scheme consists of a dedicated channel modulation and two-dimensional forward error correction. The channel modulation is based on specially designed code tables. The code tables prevent the propagation of errors if one nucleotide is in error but not more than 2 data bits will be incorrectly decoded. Today’s synthesizers and sequencers often find it challenging to correctly process oligos with series of identical nucleotides. Code tables can also be used to generate oligos with sequences of identical nucleotides that do not exceed 3. Finally, code tables can ensure that oligos with sections of reverse complementary series of nucleotides will not be generated, since such oligos are also challenging to handle.

The generated oligos are organized as virtual blocks. The blocks are protected independently from each other. The two dimensions of error correction of the blocks consists of three modules, which are illustrated in Figure 5. The modules are:

- ‘Horizontal’ error protection of the oligo addresses
- ‘Vertical’ error protection of the stored data
- ‘Horizontal’ error detection applied to the complete oligos

Together with the channel modulation, the three modules of the error correction scheme provide a novel capability, namely the ability to detect and correct insertion and deletion errors as well as substitution errors in single oligos.

4. A CODING SCHEME TO HANDLE SYNTHESIS AND SEQUENCING ERRORS

Note that DNA storage so far looks quite similar to storage on magnetic or other media (data rate, indices, etc). However, DNA storage is fundamentally different in some ways.

First, two-chemical processes typically generate not just one oligo (e.g., for a thumbprint ACGAGAAGCAAGAAGTTTAC) but hundreds of copies of that oligo. The number of available copies of an oligo is referred to as coverage. All the oligos and their hundreds of copies are produced and stored together, typically in a form of a liquid. Clearly, coverage will impact the way we handle errors created when synthesizing oligos and will determine how many copies of an error need to be sequenced when recovering the stored data.

Second, synthesizing and sequencing oligos is a process prone to errors. There are three main kinds of errors:

- Insertion error: A nucleotide is inserted by mistake into an oligo (say, an extra T or A in a chain). The oligo is then longer by one nucleotide.
- Deletion error: A nucleotide is deleted by mistake from an oligo. The oligo is then shorter by one nucleotide.
- Substitution error: A nucleotide is replaced by another nucleotide.

Insertion and deletion errors have significant impact because all nucleotides following the occurrence of such an error are shifted and therefore cannot be decoded anymore without additional information. Substitution errors in contrast only have a local impact on the final nucleotide.

In order to be able to store data efficiently and reliably in DNA molecules a dedicated error correction scheme is required, which in particular needs to handle insertion and deletion errors, and which hopefully can take advantage of the potentially very high coverage available at least for some of the oligos.

Considering one possible scheme, we conclude this section with some practical considerations. Let us assume we would like to store a short movie, which would require 1 Gbyte of data. We also assume that oligos are 200 nucleotides long, and each nucleotide can represent 2 bits of payload. Each oligo can represent 400 bits, or 50 bytes, while about 4 bytes are required to store indices, which together means a payload per oligo of 46 bytes. This in turn means that we would need to synthesize about 21.73 million oligos to achieve the movie.

Furthermore, depending on the error rate and the level of desired error correction, we may need even more oligos. For example, to store the movie at high error rate, certainly higher than the rates found in the data of Church’s DNA book data storage experiment from 2012 [3], [4]. The next step, of course, is to validate the scheme in ‘real-life’ and to measure the entire process, from Step 1 to Step 6 as described in Section 1. This is ongoing in collaboration with Church’s group and we expect to describe results in an upcoming paper.

Beyond validation, we are pursuing the work in several directions. For example, our current encoding scheme is adapted to the characteristics of today’s synthesizers and sequencers. However, rapid technological advances will surely change some of the characteristics of those machines. We expect that the emerging nanopore technology for sequencing is well suited to very long strands of DNA, and we are currently considering appropriate schemes for these.

Since the start of this project 2 years ago, archiving Hollywood movies on DNA seemed far-fetched and extremely unrealistic. There are still roadblocks ahead, such as the availability of low-cost and high-throughput synthesis of large scale DNA, but the exponential trends we described earlier and our own work (as for example described in this paper) make us now quite confident that DNA archival is in Hollywood’s medium-term future. Biology is an unlikely ally to preserve artistic masterpieces, but we are sure that DNA storage of movie or media content is actually quite natural, with specific challenges (such as errors) but exceptional benefits (robustness-free storage for millennia). In fact, we expect the benefits to be such that DNA archiving will revolutionize the storage and retrieval of valuable digital assets.

Acknowledgments

This paper describes at a high level a research project involving teams at Technicolor and at Harvard, with contributions to the overall project from a number of researchers in particular at Technicolor: Xiao-Ming Chen, Mark Crovella, Ingo Hütter, Klaus Gudde, Naveen Guda, Sabina Lazure and Nicolas Le Scouarnec; at Harvard: George Church, Brian Turczyk and Richard Terry. The work in this paper on coding was done at Technicolor. We expect other papers in the future to describe the various research contributions in detail.

5. CONCLUSION

We described in the paper at a high level the rationale for considering DNA storage and archiving of media data, and in more detail a specific scheme for handling errors that occur during the synthesis and sequencing steps of the process. A key property of the scheme is the ability to correct deletion, insertion and substitution errors in a single oligo. The scheme was simulated to correct errors in oligos up to high error rates, certainly higher than the rates found in DNA book data storage experiment from 2012 [3], [4]. The next step, of course, is to validate the scheme in “real-life” and to measure the entire process, from Step 1 to Step 6 as described in Section 1. This is ongoing in collaboration with Church’s group and we expect to describe results in an upcoming paper.

Beyond validation, we are pursuing the work in several directions. For example, our current encoding scheme is adapted to the characteristics of today’s synthesizers and sequencers. However, rapid technological advances will surely change some of the characteristics of those machines. We expect that the emerging nanopore technology for sequencing is well suited to very long strands of DNA, and we are currently considering appropriate schemes for these.

Since the start of this project 2 years ago, archiving Hollywood movies on DNA seemed far-fetched and extremely unrealistic. There are still roadblocks ahead, such as the availability of low-cost and high-throughput synthesis of large scale DNA, but the exponential trends we described earlier and our own work (as for example described in this paper) make us now quite confident that DNA archival is in Hollywood’s medium-term future. Biology is an unlikely ally to preserve artistic masterpieces, but we are sure that DNA storage of movie or media content is actually quite natural, with specific challenges (such as errors) but exceptional benefits (robustness-free storage for millennia). In fact, we expect the benefits to be such that DNA archiving will revolutionize the storage and retrieval of valuable digital assets.

Acknowledgments

This paper describes at a high level a research project involving teams at Technicolor and at Harvard, with contributions to the overall project from a number of researchers in particular at Technicolor: Xiao-Ming Chen, Mark Crovella, Ingo Hütter, Klaus Gudde, Naveen Guda, Sabina Lazure and Nicolas Le Scouarnec; at Harvard: George Church, Brian Turczyk and Richard Terry. The work in this paper on coding was done at Technicolor. We expect other papers in the future to describe the various research contributions in detail.

REFERENCES


1. INTRODUCTION

Capturing the scene’s light field has been an old interest in the field of computational photography [1, 2]. However, the recent release of hand held plenoptic cameras as Lytro has introduced the potentials of light field imaging to the mass market. By placing a microlens array between the main lens and the sensor, a plenoptic camera captures the direction of the light bundles that enter the camera, in addition to their intensity and color. Captured data is then demultiplexed to provide the light field, a matrix of horizontally and vertically aligned views from slightly different points of view over the scene. With the light fields, a number of natural applications have risen such as depth estimation [3, 4, 5] or post-capture refocusing [6]. However, the angular resolution of the plenoptic cameras comes at the price of lower spatial resolution of images. But promising super-resolution methods for plenoptic images have already been proposed [7, 3, 8, 4].

Among the state of art post-processing methods of the plenoptic data, only very few address the very first steps regarding raw data conversion: (i) demosaicking, which aims to recover the color content of the scene from the mosaicked raw data and then demultiplex to recover the views, and (ii) disparity estimation, as if it was a conventional image, wrongly mixes angular information (each pixel under a microlens corresponds to a different view). Angular information is particularly overlooked problem of demosaicking the views. The raw data conversion in such cameras is however barely studied in the literature. The goal of this paper is to study this problem in detail and to propose a generic demosaicking framework specifically designed for plenoptic data and inspired by multi-frame demosaicking approaches [13]. The work in [9], in which the raw image is demultiplexed without demosaicking, and we study how to recover the full RGB image is prone to tremendous errors. Therefore, we build on the interpolations inspired by multi-frame demosaicking and focus on the raw multiplexed data. Note that the demultiplexing step, we take the first step towards light field super-resolution with negligible computational overload.

INDEX TERMS— plenoptic camera, multi-frame demosaicking, view demultiplexing, disparity estimation.

ABSTRACT

Light field imaging has been recently introduced to mass market by the hand held plenoptic camera Lytro. Thanks to a microlens array placed between the main lens and the sensor, the captured data contains different views of the scene from different view points. This offers several post-capture applications, e.g., computationally changing the main lens focus. The raw data conversion in such cameras is however barely studied in the literature. The goal of this paper is to study the particularly overlooked problem of demosaicking the views for plenoptic cameras such as Lytro. We exploit the redundant sampling of scene content in the views, and show that disparities estimated from the mosaicked data can guide the demosaicking, resulting in minimum artifacts compared to the state of art methods. Besides, by properly addressing the view demultiplexing step, we take the first step towards light field super-resolution with negligible computational overload.

Fig. 1 - A zoom-in of view demultiplexing as in [9]. The matrix of views has as many views as there are pixels under each microlens (66 views in this example), two of which are shown on the right.

captured raw data (discussed only by [10, 7]) and (ii) view demultiplexing, which consists in reordering the pixels based on microlenses positions in order to recover the matrix of views (discussed only by [11, 12, 9] for Lytro).

Most of the works in the literature propose to first demosaic the raw data and then demultiplex to recover the views, but this leads to color artifacts on the views. By construction, neighbor pixels in a plenoptic raw image contain different angular information (each pixel under a microlens corresponds to a different view). So, demosaicking the raw plenoptic image, as if it was a conventional image, wrongly mixes angular information: classical algorithms interpolate neighbor color values, which causes the so-called view cross-talk artifacts. Besides, it has been shown in [9] that disparity estimation from views obtained from such a demosaicked raw image is prone to tremendous errors. Therefore, we build on the work in [9], in which the raw image is demultiplexed without demosaicking, and we study how to recover the full RGB views. This means that demosaicking is done on the views and not on the raw multiplexed data. Note that the demultiplexing step (pixel reordering) transforms the Bayer pattern on the raw data into a new non-global color patterns (see Fig. 1, Fig. 2(a) and [9] for more details). On these new irregular color patterns, classical demosaicking algorithms poorly recover highly textured areas. In this paper, we propose a generic demosaicking framework specifically designed for plenoptic data and inspired by multi-frame demosaicking approaches [13]. The goal is to increase the chromatic reso-