Cluster Tagging: Robust Fiducial Tracking for Smart Environments

Robert Harle and Andy Hopper

Computer Laboratory
University of Cambridge
Cambridge, UK.
{*rkh23, ah12*}*@cl.cam.ac.uk*

Abstract. Fiducial scene markers provide inexpensive vision-based location systems that are of increasing interest to the Pervasive Computing community. Already established in the Augmented Reality (AR) field, markers are cheap to print and straightforward to locate in three dimensions. When used as a component of a smart environment, however, there are issues of obscuration, insufficient camera resolution and limited numbers of unique markers.
This paper looks at the advantages of clustering multiple markers together to gain resilience to these real world problems. It treats the visual channel as an erasure channel and relevant coding schemes are applied to decode data that is distributed *across* the marker cluster using an algorithm that does not require each tag to be individually numbered. The advantages of clustering are determined to be a resilience to obscuration, more robust position and pose determination, better performance when attached to inconvenient shapes, and an ability to encode more than a database key into the environment. A real world example comparing the positioning capabilities of a cluster of tags with that of a single tag is presented. It is apparent that clustering provides a position estimate that is more robust, without requiring external definition of a co-ordinate frame using a database.

1 Introduction

Recent Computer Science research has seen an explosion of interest in location-aware systems. Such systems seek to robustly determine the position and pose of a variety of objects. The process is typically separated into three stages; determine the identity of an object; measure a quantity related to distance to one or more sensors; compute a location.

Many physical mediums have been harnessed by previous research efforts designed to locate objects, including infra-red, visible light, sound (both audible and inaudible to humans) and radio (using signals from wireless LAN, bluetooth, UWB and GSM networks, to name a few) [7]. The majority tag the objects in some manner and semantically associate the object with the tag. Active tags (with a local power source) are common since they allow greater tag-sensor distances and smarter operation. However, passive tags (which draw no local power) are preferred for their low cost and minimal maintenance. Unfortunately, present day tracking systems based on passive tags can be unreliable—the increasingly pervasive RFID [16] systems are a good example. Those tracking systems that use computer vision for tracking passive ‘fiducial’ tags are, however, more robust and reliable. The use of fiducials simplifies the general problem of tracking objects in moving images, which is notoriously complex.

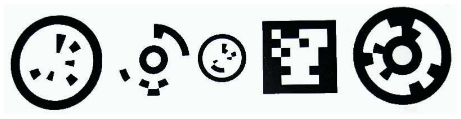


Fig. 1. A variety of tag designs that combine a recognisable shape with a unique payload

A fiducial tag is essentially a visual barcode designed to be easily recognisable to a machine. Systems based on them are well established in the postal and shipping industries (where they are used primarily for identification rather than high accuracy spatial tracking) and in Augmented Reality systems. The latter exploits the reliable identification, location and pose estimation properties of fiducial tags. Many implementations exist: the ARtoolkit [3, 4] software is widely used, but there are a growing number of competitors [6, 10, 15]. In addition, the TRIP project at Cambridge [5] used the concept for tracking objects in an office environment. The evaluation framework presented within this paper makes use of Cantag, an open source competitor to these systems, developed in-house, that is more flexible and extensible [1].

The advantages of fiducial systems are manyfold: markers can be printed quickly and cheaply, using commodity items available in every office. Even the sensor hardware amounts to off-the-shelf cameras and, when properly calibrated, it gives accurate orientation estimates (something often lacking in fine-grained location systems).

The basic operation of a fiducial system is straightforward. A camera captures an image of one or more fiducial tags. Each tag has two components: at least one recognisable geometric shape and a payload (Figure 1). The shape acts as a-priori knowledge about the tag and by searching the image for valid perspective projections of this shape we can identify where tags are in the image and calculate the position and pose that would give rise to that perspective shape. For example, ARtoolkit uses squares (becoming quadrilaterals under projection) and TRIP uses circles (becoming ellipses). Once all the tags in a scene have been determined they must be identified; this is achieved through the payload which is a per-tag symbol or code that is unique. ARtoolkit uses monochrome shapes as an identifying payload and pattern matching for recognition, although herein we adopt the binary coding found in the other systems since it lends itself to analysis. For such encoding, the payload size (in bits) is determined by the size of the tag and the size of a single bit element (the *feature size*).

In this paper we develop the idea of tagging objects with multiple tags in a manner we term ‘cluster tagging’, where:

1. Tags are used for both data communication and location information,
2. Multiple tags are used to increase resilience to obscuration,
3. Tags are arranged in a known spatial configuration,
4. Data is encoded redundantly *across* the tags,

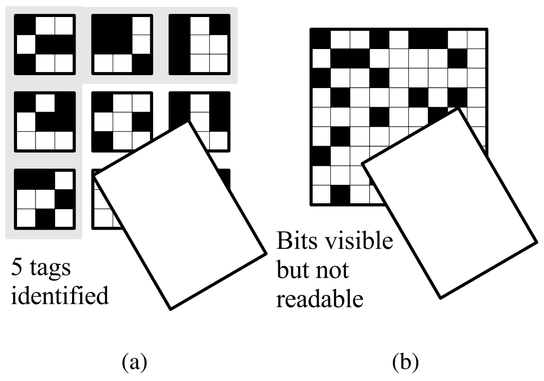


Fig. 2. Clustering advantages (a) A singly-tagged object has visible bits, but the lack of a full shape border (a square).prevents them being read (b) Clustering small tags allows some data to be read and potentially 20 corner correspondences to be identified for more accurate pose and position determination.

5. Tags are *not* uniquely indexed.

The use of multiple tags in fiducial tracking systems is not in itself novel. Both AR-toolkit and ARtag support the use of multiple tags in labelling objects. A database maps arrays of tags to known positions. When a subset of the array is observed, the system is able to compute where the other tags in the array are, providing a degree of robustness in tracking. These systems also permit the computation of camera position and pose given one or more sighted tags in an array [14]. Cluster tagging is distinguished from these systems through criteria 1, 4, and 5 above. It allows for arrays of tags to distribute information between them. In addition, we use this opportunity to more completely map out the general advantages of multiple tagging.

The remainder of this paper looks at the motivations for cluster tagging, considers how to achieve it practically (including spatial arrangements, coding schemes, and multiple independent clusters) and gives results from a real world implementation. The solution uses coding techniques from established information theory, but is unusual in its demand for small data packets, discouraging the standard solutions.

2 Motivating Cluster Tagging

Most applications of fiducial systems require the tag-camera distance to be relatively small (one or two metres). Expanding them into larger-scale pervasive systems poses interesting problems: using today's systems we require a greater number of unique tags (i.e. more payload bits) and larger feature sizes to capture images further away with sufficient resolution. Unfortunately, a bigger feature size and payload equates to bigger tags, which increases the likelihood of obscuration. Cluster tagging addresses this problem and others:

Smaller tag sizes. Distributing data across multiple tags has the potential to allow a smaller per-tag payload size. When using pure index-based tag labelling, the payload is determined by the number of unique tags required, and any error correcting coding used. When the tags are not explicitly indexed, the payload can be chosen to be smaller, permitting a greater feature size.

Redundancy. By distributing the data *redundantly* across tags, the system can cope with a proportion being obscured (Figure 2).

Geometric arrangements. Geometric arrangements and patterns of tags can convey extra information, including where other tags *should* be. This can assist the image processing algorithms.

Better Fitting. Multiple small tags can better cover an irregular object (Figure 3).

Robust pose estimation. Each tag provides independent estimates of location and orientation for the attached object (Figure 2).

More data. Multiple tags can be used to convey more information overall. Data could be encoded about the object it is attached to (spatial bounds, composition, owner, etc). This may be sufficient to remove the dependency on a local database.

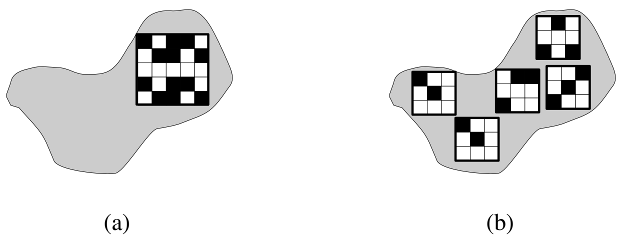


Fig. 3. Cluster tagging allows for irregular shapes. (a) Tag limited to 25 bits. (b) Clustering allows at least 45 bits

The latter idea of ‘imprinting’ data onto the environment for direct interpretation has an number of additional advantages. Data dissemination is intrinsically location private since a sensor must be physically present to read the information. This is useful to ensure physical presence rather than remote spoofing. For example, a computer may display a tag (or tags) alongside a standard login box which encodes a key that changes regularly. As a user approaches the machine, a wearable camera could decode the key and use it in conjunction with a standard password to prove both identity and physical proximity. There is also an advantage in heterogeneous deployments: tagged objects can be moved from proprietary system to proprietary system without requiring export/import of data. This is particularly important for a highly mobile user.

Disadvantages include a limit on the size of data and an inability to edit dynamic data (such as object owner). In these cases, hybrid approaches seem reasonable, where static data is encoded on tag clusters alongside a key that allows access to dynamic data.

2.1 Applications of Cluster Tagging

Applications for visual tagging that can provide identity, data and accurate location/pose information are varied. Here we present a few sample applications to clarify the contribution cluster tagging can have.

Augmented Reality. Many augmented reality systems already make use of the identity and location capabilities of single tags [2, 8, 9]. Tag sizes are typically of the order of 0.1m×0.1m and attached to stiff card to minimise warping. This size is convenient for human handling and suits the current resolution capabilities of commodity video cameras. However, uninitiated users show a tendency to obscure parts of the symbols with their hands. This problem is only compounded if the symbol is invisible to the naked eye through the use of ink pigments outside the visible range. Cluster tagging allows for this obscuration through redundancy.

Document Tracking. A complete document tracking system which can reliably and inexpensively track individual sheets of paper is the holy grail of many administerial departments. Fiducial tagging is one option for such a system: in a world of visual sensors, a system can log a timestamped location of specific documents. The inherent obscuration (Figure 4) favours small clusters of tags over larger single tags.

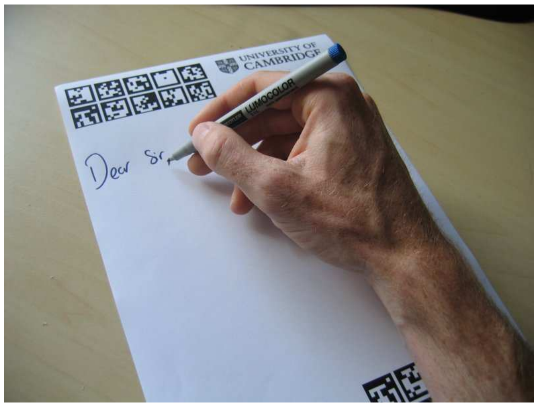
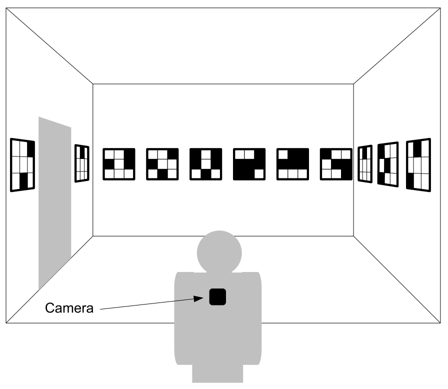
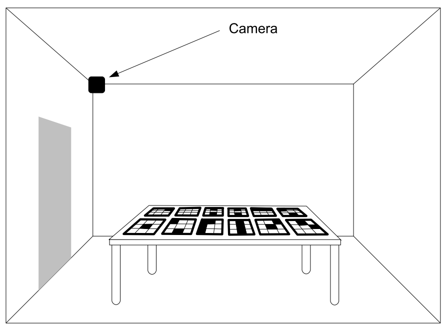


Fig. 4. Cluster tagging for document tracking

Wide Area Positioning System Fiducial tags offer the possibility of a simple positioning system: tags could be placed within an environment, acting as landmarks that a camera could compute its relative location and pose from. Cluster tagging would allow smaller tags, greater resilience to obscuration, more independent location estimates, and the possibility of imprinting room bounds and details. This concept can also be inverted, with a static camera at a known location and mobile objects cluster tagged. Such a system would be ideal for monitoring the current location and pose of large scale objects such as tables and chairs, thereby autonomously maintaining an up-to-date world model. Figure 5 illustrates the two concepts.



(a) Inside-out Positioning. Tags at known positions, camera unknown. The camera estimates its position using as many tags as possible and may use the data encoded across them to derive more information such as room name.



(b) Outside-in Positioning. Camera at known position, tags estimate table position and pose. The tags may encode details about the object as well as provide multiple estimates of position and pose and dealing with obscuration.

Fig. 5. Cluster tagging for positioning.

3 Cluster Tagging Specifics

The ideal cluster tagging scheme should aim for three major goals:

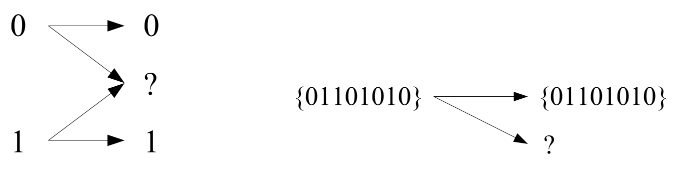
Maximal spatial coverage. The cluster should aim to maximise coverage of the associated object to increase robustness in tracking it from a variety of angles.

Minimise redundancy cost. The clustering scheme should maximise the size of the true data being encoded.

Allow cluster identification. When multiple clusters are in view it is important to be able to identify separate clusters so data does not get irretrievably mixed. The solution should avoid requiring each packet to be labelled with a unique cluster ID.

3.1 Coding Schemes

Present tag systems tend to use forward error correction (FEC) coding on a per-tag basis to ensure a valid read of the ID. This ‘inner code’ is assumed to guarantee that a given tag is decoded correctly or not at all. With the payload encoded across multiple tags, each tag carries a ‘packet’ of data and the processing of an image can be viewed as a transmission of the payload over a channel. The properties of the inner code as described characterise this channel as a packet erasure channel, where a packet is either reliably read or lost (Figure 6). Encoding data across packets such that it can be decoded over such a channel requires an effective ‘outer code’.



(a) Bit erasure channel

(b) Packet erasure channel

Fig. 6. Erasure channels.

The outer code must introduce sufficient redundancy to cope with obscuration within images (leading to erasures). This is achieved by encoding a K-packet message to N>K packets (Figure 7). Established coding schemes for erasure channels (linear codes, convolutional codes, fountain codes, etc) make one fundamental assumption: the erasures are known at the receiver. i.e. an arbitrarily chosen packet can be labelled as lost or received. This is usually achieved by indexing each packet and amortising this cost over large packet sizes. In cluster tagging, packet sizes are necessarily small and the cost of numbering each one is too high. A typical tag may only offer a payload of 25 bits (some of which are reserved to ensure a packet erasure channel)—a cluster of 20 tags would require at least 5 bits for a per tag index, significantly affecting the size of the data encoded.

Without per-packet indexing, decoding the information from a random cluster of tags would require that every possible permutation of packets and erasures be considered, each one being decoded and tested for validity (this is achievable if a CRC or similar check is included in the original data). If the coding scheme encodes to N packets, this could require N! invocations of the decoding algorithm. It is thus very important to reduce the possible permutations by exploiting the geometry of the cluster (for this reason regular tag arrangements are preferable over irregular ones since rigid structure significantly limits available permutations).

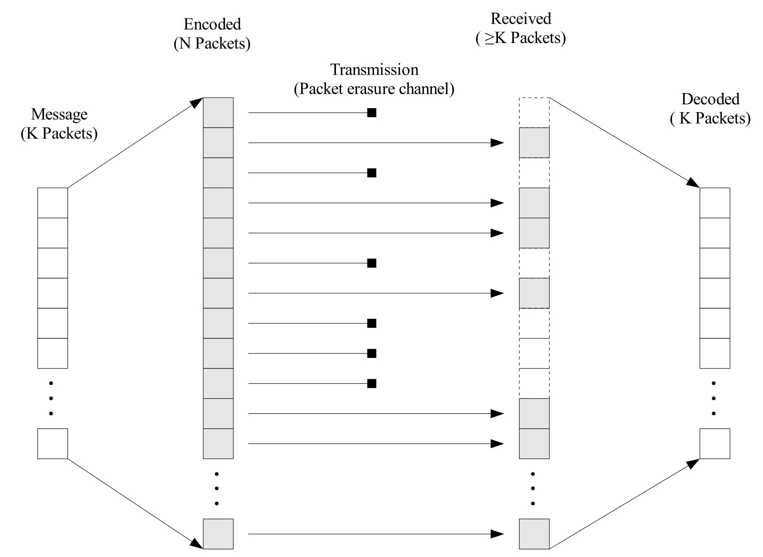


Fig. 7. Adding redundancy to deal with erasure channels.

3.2 Inferring Erasures from Structure: Relative Indexing

A cluster of tags arranged in a known configuration effectively indexes them since the configuration can provide the necessary ordering information without having to encode it explicitly in the tag payloads. However, erasures complicate this picture—when unindexed tags are missing it may not be possible to uniquely number those that remain. In these situations we select a reference tag and index the remaining tags *relative* to it. The a-priori configuration of tags should be chosen such that, given a set of tags and a reference tag, the remaining tags can be given a unique relative index. For example, consider a 3×3 array of tags that encodes 5 packets of data. The arrangement is such that the true indexes of the tags is as in Figure 8(a). Given that the tag spacing and layout is known beforehand, observing a subset of the tags (Figure 8(b)) will allow indexing relative to a member of the subset. In Figure 8(c) the reference tag is given an arbitrary *shift index*, denoted i, and the remaining four tags can then be indexed relative to it and the row width, R.

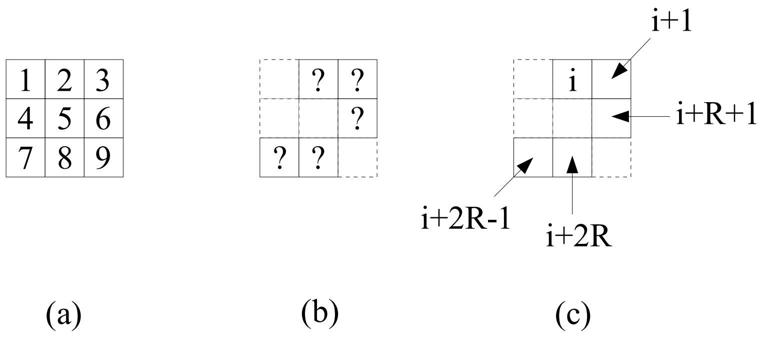


Fig. 8. Relative indexing. (a) A correctly indexed pattern as deployed (b) Five observed tags of unknown indexes (c) Indexing the first tag allows remaining tags to be relatively indexed in terms of the row width, R=3.

In this way relative indexing allows the definition of a set of possible receives/erasures (one set for each valid value of i). The difference between the largest relative-index and the reference then defines the *span*—i.e. the number of true indexes between the first and last packet decoded. For a given span, s, there may be up to (N-s) possible channel candidates, one for each of the (N-s) possibilities for the shift index, i. As a specific example, consider Figure 8 where the received vector spans 6 original indexes. This means the shift index, i, can only take one of three (=9-6) values (i=1,2, or 3) as illustrated in Figure 9. In fact, this result only applies if the layout was a row of 9 elements. The 3×3 layout can be used to further limit the possible shift indexes—since there are only three columns and there is one sighted tag either side of the first (albeit in different rows), the first tag must be in the second column; by a similar analysis we find it must be in the first row and thus this scenario is not ambiguous at all: the shift index is known instantly despite a lack of indexing within the tag payload.

In general,the worst case scenario occurs when the layout pattern is a simple row, when s=K and when the correct value of the shift index is the last one trialled. This requires (N-K) attempted decodes before the correct decode occurs. In practice, however, this is an unlikely situation, and significant computational savings will be made by considering the layout pattern carefully as described.

3.3 Determining the First Tag: Rotational Invariance

An important issue is the determination of the reference first tag. A regular arrangement of tags introduces rotational invariance which can complicate this identification—a 90° rotation of the image in Figure 8(c) results in a very different relative indexing (Figure

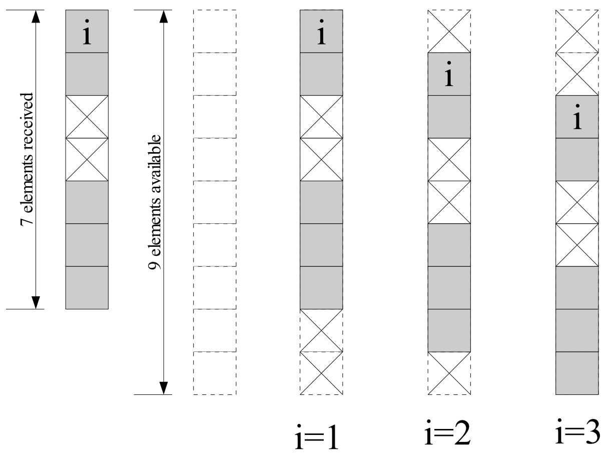


Fig. 9. Possible shifts for Figure 8.

10). Tags, then, need to incorporate directionality; a problem not unique to cluster tagging. In fact the solutions proposed for the inner coding for individual tags are equally applicable here [11]. The essential principle is to use a rotationally-invariant inner code, with specific bits used to indicate direction.

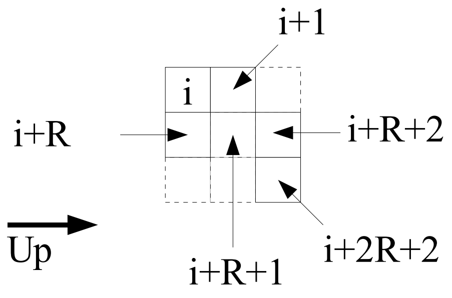
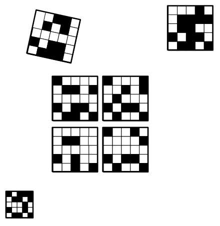


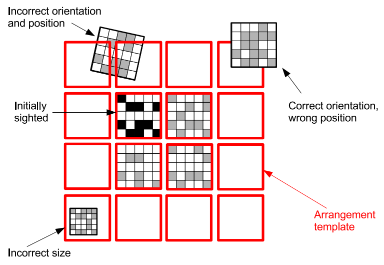
Fig. 10. Rotational invariance masks the lowest-indexed element.

4 Dealing with Multiple Clusters

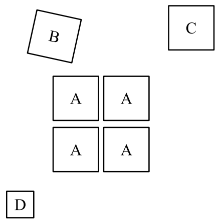
If each tag within a cluster does not carry a label identifying its parent cluster, problems can clearly arise when tags from multiple clusters are in view—not only does a viewing system need to determine the indexing of tags within a cluster, but also the clusters present. Regular geometric arrangements can also help here. Once a tag within in a cluster is sighted, the arrangement allows determination of where other tags in the same



(a) Seven tags in view



(b) Take one and derive arrangement template



(c) Tags clustered together

Fig. 11. Coping with multiple clusters in view

cluster may be and what their orientations should be, based on an arrangement template (Figure 11). In most cases, this alone should be sufficient to group tags by cluster.

A further problem may be that different geometric arrangements may suit different surface shapes, and hence multiple arrangement templates would be required. Similarly, different sizes of arrangement may also be favoured (e.g. the row width may change from cluster to cluster). These problems are very difficult to cope with robustly, so instead it is favourable to set the arrangement and its row width. This in turn allows the values of N and K to be set, simplifying decodes. The overall scale of the arrangement can, however, be varied. Thus it is possible to set an arrangement (for example, a square tessellation) and expand it to suit the object. Where whole tags do no fit on an object, they may be excluded, taken as guaranteed erasures (Figure 12).

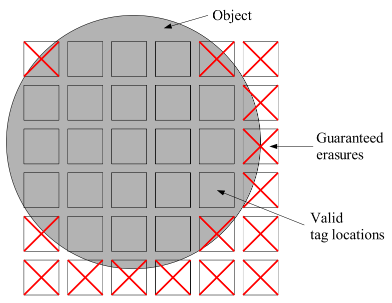


Fig. 12. Applying a standard 6×6 tag array to a non-square object with guaranteed erasures

5 Real World Results

We have implemented cluster tagging within the open-source fiducial tracking software library Cantag [1], developed in-house and freely available. Cantag supports arbitrary tag shapes, data encodings and interpretation algorithms and is well suited to both production and development of novel fiducial systems.

In the example provided here, we use a 7×7 cluster of 5-element square tags (Figure 13) to encode a global co-ordinate system by defining the position and pose of the cluster itself in that frame. A camera then decodes this information and can compute its pose in that global co-ordinate frame without an external database.

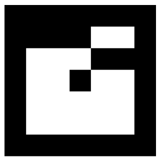


Fig. 13. A sample tag used in evaluation

The choice of a square tag allowed for ease of tessellation. Each tag carried 24 bits of data split into four blocks of 6 bits. Each block was encoded on the tag such as to readable in any orientation. One bit per block was reserved for orientation information, three bits for data and the remaining two bits formed parity check bits. This allowed each tag to represent 12 bits of data and to have sufficient resilience to produce the necessary erasure channel.

The data encoded across the tags defined three points in 3D. The first defined the position of the first tag in the cluster, the remaining two defined the direction of the x- and y- axes in the same frame. 32 bit floating point values were used for each of the nine co-ordinates, making a total data length of 288 bits. To this a 32-bit CRC was added to assist in decoding the cluster, making 310 bits to encode. This data size requires at least 26 tags (312 bits) and hence K=26, whilst N=49. Thus the cluster can cope with any 23 tags being obscured without effect.

5.1 The Outer Code

For the outer code a linear BCH code was implemented, similar to that of Rizzo [13]. The premise of such a code is that a K$\times$N generator matrix, $\mathbf{G}$, acts on an input vector,$\mathbf{x}$, of size K to produce an encoded vector of size N,

$$\mathbf{y} = \mathbf{G}\mathbf{x}. \tag{1}$$

The received data vector, $\mathbf{y}'$, is then a subset of $\mathbf{y}$ and must have a size of at least K for decode. A decode matrix can be created by first constructing a K$\times$K matrix $\mathbf{G}'$ and then inverting it.

$$\mathbf{x} = \mathbf{G}'^{-1}\mathbf{y}'. \tag{2}$$

From this relation is should be apparent that $\mathbf{G}'$ is constructed from the rows of $\mathbf{G}$ corresponding to any K elements of $\mathbf{y}'$. The remaining question is how to form the matrix $\mathbf{G}$. For this implementation, a systematic code was chosen, whereby the first K rows are the identity matrix. The remaining rows must be linearly independent, and this can be assured by the use of a Vandermonde matrix, making $G_{ij} = \alpha^{(i-1)(j-1)}$ for j>K and α the generator element for a Galois Field of the chosen size (see [13] for more details).

5.2 Positioning

Positioning of square tags in Cantag is generally based on the identification of the corners within the image. These form correspondences between the image and the real world and are sufficient to locate a tag within the reference frame of the camera. Cantag implements a variety of algorithms that have been evaluated elsewhere [12]. The key point regarding the localisation of a particular tag is that the algorithms trade-off between accuracy and computational complexity. The fastest algorithm in Cantag is presently the *Linear Projective* algorithm, based on a linearisation of the localisation problem, whilst the most accurate algorithm, *Space Search*, uses a minimisation algorithm to iteratively find the most likely tag position and pose ([12]).

In implementing cluster tagging, we use the fast linear algorithm to identify and decode as many tags as possible in the frame. The decoded data is then used to create correspondences between image pixels and the global reference frame (four correspondences per square). A minimisation algorithm is then used to find the optimal camera position, effectively inverting the Space Search algorithm.Essentially, there are six parameters necessary to define the 3D position and pose of the camera. With each iteration of the algorithm, the location of each tag corner is computed, assuming the camera to be at the present estimate. This location is then projected to an image frame, and the error distance to the corresponding projections in the true image are calculated. By minimising this error, the camera position and pose is determined. The source code for this process is contained within the present Cantag distribution as the *EstimateTransform* algorithm.

5.3 Results

The cluster was printed on A4 paper using a standard laser printer. Each tag measured 2.15cm$\times$2.15cm and the layout used row and column gaps of 0.5cm. To aid comparisons a single tag with the dimensions of the entire cluster was printed onto a second piece of paper.

In each experiment the relevant sheet was affixed to a desk and a camera mounted above it, pointing towards it. Care was taken to level the camera to point perpendicular to the desk surface. The camera position was measured to within an estimated error of 1cm in each dimension, and was not moved throughout the experiments. 100 image frames were captured with the cluster sheet in place and 100 further with the single tag sheet in the same place, suitably aligned. Using Cantag the camera position was estimated for each image. The positions were calculated in up to three different ways:

1. Using all observed tags from the cluster sheet,
2. Using each individual tag observed from the cluster sheet,
3. Using the single large tag from the single tag sheet.

The position distributions are shown in Figure 14. The estimated camera position is shown in each, although it should be noted that the experimental error in measuring this quantity is significant on this scale. For clarity, Figure 15 shows the CDFs for each of these results. Given the relatively large estimated error of the true position, we find little difference between using a cluster and a single large tag of equivalent dimensions, both of which exhibit an error consistent with today's fiducial systems.

Whilst the small position distribution for the single large tag appears to imply a better localisation than the other methods, this is not necessarily the case. Because its four corners lie away from the centre of the image, towards the bounds, they are more affected by lens distortion. Whilst we have corrected for this distortion using a standard radial model within Cantag, this is only a generic lens model and less trust can be placed in points further from the principal point in the image. The (slightly) larger spread of positions when using the entire cluster is indicative of the fact that the location algorithm has many points extracted from across the image upon which to base its estimate. The error distribution thus better reflects the true state of affairs.

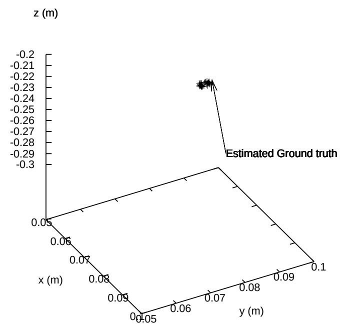
Figure 14(d) shows the results of repeating the cluster experiment with approximately one third of the cluster occluded. The data remains decodable and the position distribution changes little, as evidenced by the CDF in Figure 15. When a similar occlusion was applied to the single large tag, no positions were possible since no tags were identifiable. We conclude from this that cluster tagging has the potential to locate a camera to at least a comparable precision to that when using a single large tag. This is true even with occlusion present, increasing the robustness of the position measurement over the latter approach.

These results reflect our general findings with clustering: position estimates can be significantly improved over tags of similar size, whilst resilience is inherent in the redundancy. Beyond the example given here, the code for camera position and pose estimation has been shown to be robust and reliable and many empirical experiments using different arrays of tags have demonstrated the strengths of the approach over single tags.

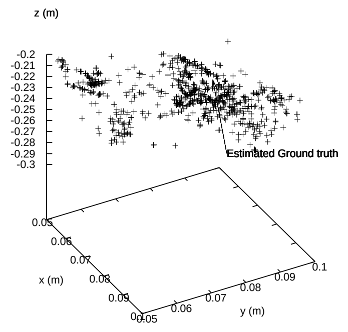
6 Advantages and Disadvantages

The ideal positioning solution depends heavily on the application envisaged. Our experiences have found cluster tagging to be particularly useful for ubiquitous computing. Lighting conditions can be a particular problem with single tag systems in a general environment, and cluster tagging assists by increasing the chance that a useful proportion of encoded data can be retrieved by using smaller tags.

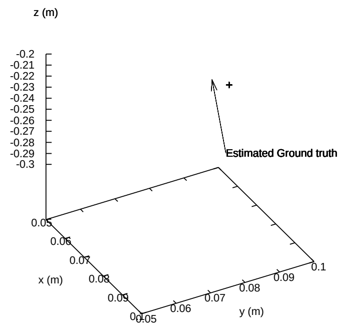
In terms of disadvantages, there is a clear problem with how to 'print' the information to a space. Traditional inks will fade and the initial deployment is complex unless



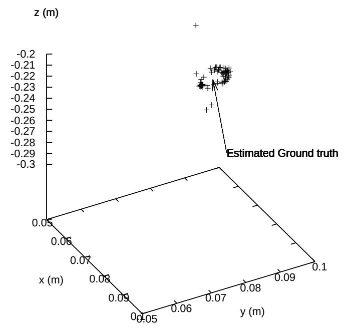
(a) Using entire cluster



(b) Using individual tags



(c) Using a single large tag



(d) Occluding the cluster

Fig. 14. Position distributions for different processing and setups

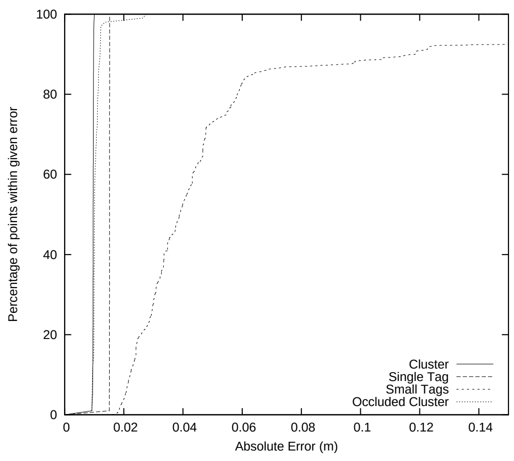


Fig. 15. Cumulative density functions of results in Figure 14

the tags are incorporated a the time of manufacture (for example into wallpaper). The difficulty in accurately placing tags both in a global reference frame and within a local geometric template during a very large and ubiquitous deployment may counteract the location accuracy advantages on offer. Nonetheless, we expect cluster tagging to perform as well or better than single tagging in almost every respect in the general case.

7 Future Work

Cluster tagging has associated with it similar questions to single tagging: what is the best tag shape? What bit error rate can be expected? How does resolution affect the result? These questions are difficult to answer, but the Cantag platform provides the flexibility to investigate them.

As regards cluster tagging itself, there are many open questions regarding the optimal coding scheme, the best way to optimise the decode process and the best way to combine all the redundant information to produce the best estimates of position and pose; we hope to investigate the many options available and also to move the symbols from the visible spectrum into the near infrared spectrum to minimise aesthetic disturbance.

8 Conclusions

This paper has introduced the concept of cluster tagging with fiducial tags. Cluster tagging uses multiple small tags to encode data redundantly across a space, such that observing any subset is sufficient to recover the data fully. The advantages to doing this over a traditional single-tag deployment have been determined as a reduced need for a database, inherent location privacy, greater robustness for tracking, the capability to cope with partial view occlusion, and to make better use of the available tagging space. We have implemented and analyzed the technique in the real world and hope to develop and deploy the idea in the future.

9 Acknowledgements

The authors would like to thank the reviewers who have helped to improve the work presented in this paper, and to recognise the contributions of the other Cantag authors.

References

1. A. Rice, A. R. Beresford, R. K. Harle. Cantag: an Open Source Software Toolkit for Designing and Deploying Marker-based Vision Systems In *Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computer and Communications (PerCom), Pisa, Italy, 13-17 Mar*, 2006.
2. M. Billinghurst, H.Kato, and I. Poupyrev. The MagicBook: Moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, pages 2–4, May 2001.
3. Mark Billinghurst, Hirkazu Kato, and Ivan Poupyrev. The MagicBook—moving seamlessly between reality and virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, 2001.
4. Mark Billinghurst and Hirokazu Kato. Collaborative mixed reality. In *Proceedings of the First International Symposium on Mixed Reality*, pages 261–284, 1999.
5. Diego López de Ipiña, Paulo R. S. Mendonạ, and Andy Hopper. TRIP: a low-cost vision-based location system for ubiquitous computing. *Personal and Ubiquitous Computing*, 6(3):206–219, May 2002.
6. Mark Fiala. ARTag revision 1, a fiducial marker system using digital techniques. Technical report, National Research Council Canada, 2004.
7. J. Hightower and G. Borriello. Location sensing techniques. *IEEE Computer*, August 2001.
8. H.Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, and K. Tachibana. Virtual object manipulation on a table-top AR environment. In *Proceedings of ISAR 2000*, October 2000.
9. Hideki Koike, Yoichi Sato, and Yoshinori Kobayashi. Integrating paper and digital information on enhanceddesk: a method for realtime finger tracking on an augmented desk system. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8(4):307–322, December 2001.
10. Jun Rekimoto. Matrix: A realtime object identification and registration method for augmented reality. In *Proceedings of Asia Pacific Computer Human Interaction*, pages 63–68, July 1998.
11. Andrew Rice, Christopher Cain, and John Fawcett. Dependable coding for fiducial tags. In *Proceedings of the 2nd Ubiquitous Computing Symposium*, pages 155–163, 2004.
12. Andrew Rice and Robert Harle. Evaluating lateration-based positioning algorithms for fine-grained tracking. In *Proceedings of the Joint Workshop on Foundations of Mobile Computing (DIAL-M-POMC) 2005*. ACM Press, 2005.
13. L. Rizzo. Effective Erasure Codes for Reliable Computer Communication Protocols. *ACM Computer Communications*, 27(2):24–36, April 1997.
14. U. Neumann and S. You and Y. Cho and Lee and J. Park. Augmented Reality Tracking in Natural Environments. In Y. Ohta and H. Tamura, editor, *Mixed Reality - Merging Real and Virtual Worlds*, pages 101–130. Ohmsha and Springer-Verlag, 1999.
15. Daniel Wagner, Thomas Pintaric, Florian Ledermann, and Dieter Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Third International Conference on Pervasive Computing*, 2005.

16. Roy Want. RFID: A key to automating everything. *Scientific American*, pages 56–65, January 2003.