

# Chapter 12

## Vascular Biometric Graph Comparison: Theory and Performance



Arathi Arakala, Stephen Davis and K. J. Horadam

**Abstract** Vascular biometric templates are gaining increasing popularity due to simple and contact free capture and resilience to presentation attacks. We present the state of the art in Biometric Graph Comparison, a technique to register and compare vascular biometric templates by representing them as formal graphs. Such graphs consist of a set of vertices, representing the branch, termination and crossover points in the vascular pattern, and a set of edges. An edge represents the relationship between a pair of feature points that are directly connected by a vessel segment in a vascular biometric image. We summarise how this information has been successfully used over the past 8 years to improve registration and recognition performance for the vasculature under the palm, wrist, hand and retina. The structural properties of biometric graphs from these modalities differ, with retina graphs having the largest number of vertices on average and the most complex structure, and hand graphs having the smallest number of vertices on average and being the least connected. All vascular graphs have similarities to trees, with the ratio of edges to vertices being close to 1. We describe our most recent algorithms for biometric graph registration and comparison, and our performance results. We are interested in the possibility of using biometric graphs in a template protection scheme based on the paradigm of dissimilarity vectors. As a first step, we wish to improve registration. Certain modalities like retina have an intrinsic reference frame that makes registration more straightforward. Other modalities may not have an intrinsic reference frame. To overcome this, we introduce the notion of anchors—subgraphs of a biometric graph, having between 5 and 10 vertices, that occur consistently in samples from the same individual—that would enable the dissimilarity vector scheme to be applied to any vascular modality. Experiments on palm and wrist databases show that all individuals had at least some sets of 6 captures which could be used to identify an anchor,

---

A. Arakala (✉) · S. Davis · K. J. Horadam

Discipline of Mathematical Sciences, School of Science, RMIT University, Melbourne, Australia  
e-mail: [arathi.arakala@rmit.edu.au](mailto:arathi.arakala@rmit.edu.au)

S. Davis

e-mail: [stephen.davis@rmit.edu.au](mailto:stephen.davis@rmit.edu.au)

K. J. Horadam

e-mail: [kathy.horadam@rmit.edu.au](mailto:kathy.horadam@rmit.edu.au)

© The Author(s) 2020

A. Uhl et al. (eds.), *Handbook of Vascular Biometrics*, Advances in Computer Vision and Pattern Recognition, [https://doi.org/10.1007/978-3-030-27731-4\\_12](https://doi.org/10.1007/978-3-030-27731-4_12)

355

and anchors were identified in 94% and 88% for the palm and wrist databases, respectively.

**Keywords** Biometric graphs · Graph comparison · Dissimilarity vector representation · Vascular graphs

## 12.1 Introduction

The purpose of this Chapter is to provide a single resource for biometric researchers to learn and use the current state of the art in Biometric Graph Comparison<sup>1</sup> for vascular modalities.

Vascular biometric recognition is the process of identifying and verifying an individual using the intricate vascular pattern in the body. Sources of vascular patterns for personal identification and verification are the palm, dorsal hand, wrist, retina, finger and face. Traditionally, vascular patterns have been compared using feature-based or image-based templates. Here we work with feature-based templates only. The basic feature points in a vascular network are vessel terminations (where the vessels leave the image frame of reference or become too fine to be captured in the image), vessel bifurcations (where one vessel splits into two) or (in two-dimensional images) vessel crossovers, where two vessels appear to intersect.

Biometric Graph Comparison (BGC) is a feature-based process, which enhances and improves on traditional point pattern matching methods for many vascular modalities. Its key idea is the replacement of a feature point based representation of a biometric image by a spatial graph based representation, where the graph edges provide a formal and concise representation of the vessel segments between feature points, thus incorporating connectivity of feature points into the biometric template. This added dimension makes the concepts and techniques of graph theory newly available to vascular biometric identification and verification.

In particular, the comparison process is treated as a noisy graph comparison problem, involving local minimisation of a graph editing algorithm. From this, we can extract a Maximum Common Subgraph (MCS), the noisily matched part found to be common to the two graphs being compared. Part of the fascination and value of working with BGC has been to investigate the topology of the MCS: MCSs from two vascular images from the same biometric instance usually look very different from those from different instances.

Over the years since its introduction, BGC has been shown by ourselves and colleagues to improve recognition accuracy, and if more of the topology of the MCS is used to discriminate between genuine and impostor comparisons, this improvement can be quite dramatic. It is also possible to exploit specific graphical characteristics of different modalities to speed up the recognition process.

---

<sup>1</sup>Previously we used the non-standard term Biometric Graph Matching (BGM).

The Chapter is organised as follows. In Sect. 12.2, we define the vascular Biometric Graph and explain its background and context. A very brief description is given of its extraction from a vascular image. Section 12.3 outlines the formal description of the two components, registration and comparison, of BGC, with some history of its development from its earliest form in [7] to its newest form presented here. (Pseudocode for our Algorithms appears in the Appendix.) In Sect. 12.4, we summarise the body of results in [6–8, 20, 21]. We compare the graph topology of the public retina, hand, palm and wrist databases we use, and describe the topological features of MCSs we have identified from which to derive comparison scores. We provide the supporting evidence for our view that the Biometric Graph representation increases the speed and accuracy of registration, accuracy of comparison, and that using multiple graph structures in the MCS can improve comparison scores over single structures.

Section 12.5 presents one stage of an application of BGC to the problem of privacy protection of vascular templates. The key idea is a feature transformation using a dissimilarity vector approach. Preliminary investigation of the comparison performance of this approach has given encouraging results for retina databases, where an intrinsic alignment exists in the images [5]. A new problem is faced if no such alignment exists. Here we present our first results on a potential solution to this problem, where we look for small but characteristic structures we call “anchors”, which appear in sufficiently many of an individual’s samples to be used for registration.

## 12.2 The Biometric Graph

This section presents the Biometric Graph we use for application to vascular biometric modalities. We describe our motivation for using a spatial graph representation over more traditional feature point based templates. We provide a formal definition of a vascular Biometric Graph and give a brief overview of the extraction process.

### 12.2.1 *The Biometric Graph*

Biometric Graphs, as we define them, were first introduced in 2011 [17] for the fingerprint modality. Extraction of ridge bifurcations and terminations as feature points is a fundamental technique in a ridge-based modality, and usually, ridge skeletons are also extracted from images. The novelty of the Biometric Graph concept lies in constructing a formal spatial graph from these extracted feature points only. Each feature point is represented as a vertex (also called a node). An edge (also called a link) is a straight line drawn between adjacent pairs of feature points on the skeleton. The edge preserves, in summary form, the connectivity relationship between feature points typically found by tracing along the ridge skeleton. (This differs from the earlier ISO/IEC 19794–8:2006 standard, in which additional “virtual minutiae” and

“continuation minutiae” are inserted along the skeleton, to facilitate piecewise-linear representation of the connecting ridgeline.) A disadvantage of our representation is that more detailed information held by a ridgeline curving between feature points is lost, particularly in regions of high curvature where an edge forms a shortcut between feature points. Figure 12.9 in Appendix 1 demonstrates this. An advantage of our spatial graph representation which can outweigh this loss of information is computational efficiency. An edge can be represented in code concisely by its two end vertices. Furthermore, the full repertoire of graph theoretical techniques is available for data analysis.

### 12.2.1.1 Vascular Graphs

Direct observation of two-dimensional images of vessel-based modalities shows the physical branching and crossing network of vessels strongly resembles a formal spatial graph drawn in the plane. For example, there is some visible similarity between the pattern of the principal retinal vessels and a rooted tree (with the root vertex in the optic disc), and some visible similarity between the pattern of the principal wrist vessels and a ladder graph or lattice. These similarities to spatial graphs are more pronounced to the naked eye for vascular modalities than in the ridge-based modalities for which we first studied Biometric Graphs. Fundamentally, this is because blood vessels do not often exhibit high curvature, so in most cases the vessel segment between adjacent feature points is quite well represented by a straight line. This was our motivation in [7] for introducing Biometric Graphs and Biometric Graph Comparison into vascular biometric modalities.

The idea of a vascular graph has arisen independently (and at approximately the same time) in the biomedical literature. Drechsler and Laura [13], working with three-dimensional hepatic vessel CT (computed tomography) images of the liver, extract a three-dimensional vascular graph from the vessel skeleton (using voxels not pixels—crossovers do not occur). They classify voxels into three classes: regular, end (terminations) and branch (bifurcations). Branch and end voxels are represented by vertices in the graph, while regular voxels are grouped and represented by edges. The vascular graph provides data for further image recognition, registration and surgical planning. Deng et al. [12] extract a vascular graph (which they term a vascular structure graph model) from the skeleton of the vessel tree in two-dimensional retinal fundus images, to register the images for clinical diagnosis and treatment of retina diseases.

**Definition 12.1** A *vascular graph* extracted from a vascular image is a spatial graph with the vessel features of terminations and bifurcations (and crossovers if the image is two-dimensional) forming the graph vertices. A pair of vertices will have an edge between them if and only if we can trace along a vessel from one feature to another, without encountering any other feature in between. More formally, if  $I$  is a vascular image then its vascular graph is  $g = (V, E, \mu, \nu, A)$ , where  $V$  is a set of vertices representing the feature points extracted from  $I$ ,  $E$  is a set of edges between those

pairs of vertices representing feature points which are adjacent in  $I$ ,  $\mu$  is the vertex labelling function,  $\nu$  is the edge labelling function and  $A$  is the attribute set (which may be empty) comprising a set of vascular attributes that apply to feature points or to the vessel segments connecting them. The *order* of  $g$  is the number of vertices  $|V|$  and the *size* of  $g$  is the number of edges  $|E|$ . If the vascular image  $I$  is of a biometric modality then  $g$  is a (*vascular*) *Biometric Graph (BG)*.

For the BGs in our research,  $\mu$  associates each vertex with its unique two-dimensional spatial coordinates  $(x, y)$  while  $\nu$  associates each edge with its two-dimensional Euclidean length  $\ell$  and slope  $\theta$ .

## 12.2.2 Biometric Graph Extraction

To construct the Biometric Graph from a two-dimensional biometric image, the vessel skeleton is extracted from the image and the feature points are found. The feature points are labelled to form the vertex set, and their coordinates are recorded. The existence of an edge between vertices is determined by tracing the skeleton from each feature point until another is encountered. The length and slope of each edge is calculated and recorded. Other feature point and vessel segment attributes can be calculated at the same time.

Differences in image capture device and lighting source require different image processing techniques for different modalities to reduce noise. There are some common image processing steps in skeleton extraction for any vascular modality, including grayscale conversion, Region-of-Interest (ROI) selection, noise reduction, binarisation and skeleton thinning. Those we employed for palm, dorsal hand, wrist and retina images are described in [6, 8, 20, 21] and the references therein, and will not be further detailed here. For skeleton extraction from finger images, see [23].

A specific problem encountered with extracted skeletons has been the existence of genuine short spurs due to tiny vessels and spurious short spurs due to noise [6, 8, 13, 23]. This is overcome in post-processing by pruning the skeleton of branches shorter than a heuristically selected threshold such as 5, 10 or 15 pixels. For palm vessels, an additional complication has been the inclusion of short to medium length spurs in the skeleton which correspond to skin ridges or flexion creases. Palm principal ridges and creases can be considered as part of the biometric pattern and are difficult to remove completely. However, our experiments have shown that removing the short to medium spurs after the detection of vertices and edges improves the process of registration and comparison. See [8] for details. Wrist vessel skeletons often have segments running perpendicular to the main direction of the vessels, some of which are due to flexion creases, but as some are vessels, these segments are not removed [6].

Feature points are extracted from the 1-pixel-wide skeleton by counting neighbouring pixels in a standard  $3 \times 3$  pixel window moving across the skeleton. One neighbour indicates a termination pixel, two neighbours indicate a vessel pixel, three neighbours indicate a bifurcation pixel and four or more neighbours indicate a crossover pixel. As a consequence of image noise, neighbouring pixels in the same  $3 \times 3$  pixel region may be labelled as bifurcation points. To handle this, if a central

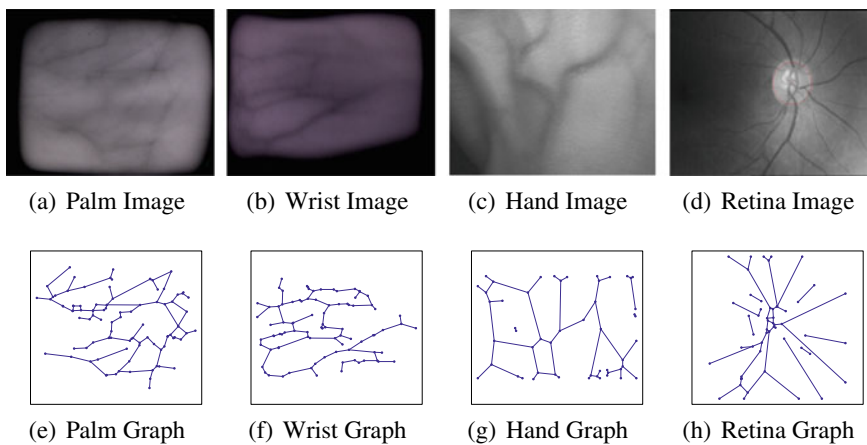
pixel is a bifurcation point and there are two or more neighbours which are bifurcation points on different sides of the central pixel, then only the central pixel is listed as the bifurcation point.

A much faster method of extracting feature points from the vessel skeleton, which may be preferable to the above, is the use of convolutional kernels as in [1].

The vertex and edge labels form the basic biometric template. Additional attributes can be extracted from the skeleton to create richer templates. Vertex attributes can include type (termination, branching or crossover). Edge attributes can include the length (as a pixel count) of the skeleton segment between two feature points and the vessel segment average width (or calibre) which can be measured before thinning the skeleton.

Figure 12.1 shows typical vascular pattern images from the databases of each of the four modalities we have investigated and their corresponding Biometric Graphs, extracted as above.

Biometric Graphs have been similarly extracted from skeletons of finger vessels by Nibbelke [23] and from skeletons of face vessels by Gouru [16]. Whilst skeleton tracing is probably the best technique in current use for identifying adjacent feature points in the image skeleton, it is possible that alternatives may prove useful. Khakzar and Pourghassem [19], working with retina images, determine for each pair of feature points whether they are adjacent or not by deleting the two points from the skeleton and checking if the remaining connected components of the skeleton all contain feature points. Existence of a component without feature points means the two points are connected in the skeleton, otherwise they are not. Connectivity is recorded in (the upper half of) an adjacency matrix. However, edge attributes aren't extracted in this approach, and since the adjacency matrix can be found immediately from the edges found by skeleton tracing, it is not clear if the approach has advantages over skeleton tracing.



**Fig. 12.1** Vascular patterns from four modalities **a** Palm **b** Wrist **c** Hand and **d** Retina vessels and their corresponding spatial graphs in **(e-h)**

## 12.3 The Biometric Graph Comparison Algorithm

In this section, we present a formal description of the Biometric Graph Comparison Algorithm. The algorithm has two parts: BGR (Registration) which requires 4 steps; and BGC (Comparison), in which the 3 steps are finding the graph edit distance, identifying the Maximum Common Subgraph (MCS) and scoring comparisons using graph-based difference measures.

In our opinion, graph registration is the key component of the algorithm, and is more critical than the graph comparison component. Although it can often be assumed that the capture mechanism enforces an approximate alignment of biometric images in the first place, experience tells us that alignment is seldom ideal, and large differences can occur between captures from the same person, particularly as the time between captures increases. Unless two extracted BGs from the same biometric instance can be aligned well, comparison cannot be effective. Essentially this is because we need a good similarity score for a genuine match, in order to minimise the number of false non-matches. The variance of genuine similarity scores across a population tends to be higher than the variance of impostor similarity scores, which have a distribution of low scores that is roughly independent of registration.

Alignment on a point pattern, such as the set of vertices in a BG, is a standard matching technique. Commonly used methods are the Iterative Closest Point (ICP) algorithm and the Modified Hausdorff Distance (MHD) algorithm. Registration using point pattern alignment algorithms has been previously studied for hand and palm vasculature. In 2009, Chen et al. [10] showed that ICP provided better alignment and consequently superior recognition results than either MHD or point-to-point comparison for palm veins.

In 2014, we showed [21] that for hand veins, registering on edges of BGs using our Biometric Graph Registration (BGR) algorithm gives as good or better recognition performance than either ICP or MHD applied to the point patterns of vertices, especially when the BGs are small. Subsequently, we have modified BGR to permit registration on structures larger than single edges.

### 12.3.1 BGR-Biometric Graph Registration

Our registration algorithm, in essence, a greedy RANSAC algorithm, looks for structural similarities in a pair of graphs on which to align them, so that the two graphs are in the same spatial frame, free from the effects of translation and rotation of their images during capture.

There is no restriction on what type of structure (i.e. subgraph) can be used for alignment within a particular modality and database. For instance, the algorithm could be tested on a database for different choices of alignment structure, so that the structure giving the best performance could be identified. Or, the frequency of occurrence of different types of structure within the database could be used to select

a preferred structure. Or, if a particular structure was found to be characteristic of a database, appearing more frequently than might be expected in a random spatial graph with comparable numbers of vertices and edges such a “motif” structure could be identified and chosen to align on. Or, it is possible that for a particular modality, each biometric instance exhibits a characteristic structure in most of its images, and such an “anchor” structure could be used for registration.

If the modality possesses an intrinsic coordinate system which can be identified in each database image, registration by the structure might not be required.

To take advantage of the additional structural information in a BG, we align on an edge, or a more complex subgraph such as a *claw*<sup>2</sup> (a degree 3 vertex plus its 3 adjacent edges and 3 neighbouring vertices), a pair of claws joined by a common edge (which we call a *two-claw*), or we could choose a cycle of length 3 or 4. In theory there is no restriction to the type of subgraph chosen for alignment, but computational limits, time constraints and the smaller number of more complex structures present in a BG usually dictate that simpler structures are preferable.

The BGR algorithm is described in more detail in Appendix 2. The algorithm is flexible so that any structure could be used for alignment. It has four steps which are outlined in the following subsection. The four design parameters in the BGR algorithm are a *structure*  $S$ , a *similarity score function*  $f$  depending on the structure selected, a structure pair *shortlist length*  $L$  and a vertex comparison *tolerance*  $\varepsilon$ . The structures  $S$  we have used are: *Edges* ( $E$ ), *Claws* ( $C$ ) and *Two-claws* ( $T$ ). If we need to specify the parameters we denote the algorithm by  $\text{BGR}(S, f, L, \varepsilon)$ .

Our initial implementation of BGR in 2011 was for  $\text{BGR}(E, f, L, \varepsilon)$  [7]. This has undergone some modification in the intervening years, so that in 2015 we introduced an improved shortlisting mechanism [8] for edge pairs in Step 3 of BGR rather than simply selecting the  $L$  highest scoring pairs. We discovered that most edge pairs (in palm BGs) were short and often scored a high rank compared to longer pairs. This prevented longer pairs that gave a better registration from appearing in the top  $L$  shortlist. To overcome this, for  $\text{BGR}(E, f, L, \varepsilon)$  we split the set of edge pairs into long and short edge pairs. The mean of the medians of the edge lengths in the two graphs is selected as the threshold. If both edges of an edge pair are longer than this threshold, the edge pair is categorised as long. All other edge pairs are labelled as short. The shortlist consists of the  $L/2$  top scoring long edge pairs and the  $L/2$  top scoring short edge pairs. This modification ensures that long edge pairs that potentially give better alignment can be included in the shortlist to get a better registration of the graphs. This modification implies that lines 13–19 in the general algorithm in Appendix 2 are run twice, once each for the  $L/2$  long and  $L/2$  short edges.

In our earlier work [5–8, 20, 21] we assumed that the images in a database are roughly pre-aligned. Here, to provide the most generally applicable registration algorithm, we have modified the similarity scoring of edge pairs in Step 2 of BGR to remove any dependence on pre-alignment. This modification means that in lines

---

<sup>2</sup>Previously we called this a star, inaccurately, as it is formally a 3-star: an  $n$ -star is a vertex of degree  $n \geq 1$ , plus its adjacent edges and neighbouring vertices.



29–31 of the algorithm in Appendix 2, only the edge lengths are used and edge slopes are not.

### 12.3.1.1 BGR Algorithm Outline

**Step 1: Initialisation** Select  $S$ ,  $f$ ,  $L$  and  $\varepsilon$ . The two graphs  $g$  and  $g'$  to be registered are inputs to the algorithm. The registration process begins by identifying and listing all the structures of the selected type  $S$  in each graph.

**Step 2: Similarity scoring structure pairs** Each structure in the first graph  $g$  and structure in the second graph  $g'$  is compared using  $f$  to obtain a similarity score. The similarity function chosen depends on the structure. For example, when edge pairs are compared they are scored based on the similarity of their lengths only (if no pre-alignment is assumed) or of their lengths and slopes (if some pre-alignment is assumed). When claw pairs are compared they are scored based on the similarity of the lengths of their three edges and two included angles. When two-claw pairs are compared, the similarity of the corresponding claw structures and connecting edges determines the score.

**Step 3: Shortlisting structure pairs and aligning on them** The structure pairs are ordered based on decreasing order of similarity score. The top  $L$  high scoring structure pairs (for  $S = C$  or  $S = T$ ) or the top  $L/2$  short and top  $L/2$  long edges (for  $S = E$ ) are shortlisted for further processing. For every shortlisted structure pair, the two graphs are translated and rotated so that a specific part of the structure becomes the origin of the reference frame. For example, if edges are used, the vertex with smaller  $x$  coordinate becomes the centre of the coordinate system and the other vertex defines the direction of the positive  $x$ -axis. If claws are used, the centre of the claw becomes the origin while the longest edge defines the direction of the positive  $x$ -axis. If two-claws are used, the connecting edge defines the coordinate system, again taking the vertex with smaller  $x$  coordinate as the origin of the reference frame.

**Step 4: Pair alignment scoring and graph registration** With both graphs in the same coordinate system, aligned on a shortlisted pair, each vertex in the first graph  $g$  is matched to a vertex in the second graph  $g'$  by finding the first vertex in  $g'$  that is within  $\varepsilon$  pixels from it. If a vertex in  $g$  does not find a corresponding vertex in  $g'$  within  $\varepsilon$  pixels of it, it will not be matched. The total number of matched vertices is normalized by the geometric mean of the number of vertices in the two graphs to provide a rough measure of alignment we call *QuickScore* (QS). That is, if  $g$  has  $n$  vertices,  $g'$  has  $n'$  vertices and the aligned graphs have  $c$  matched vertices within tolerance  $\varepsilon$ , the distance between  $g$  and  $g'$  is calculated to be

$$QS(g, g') = 1 - \frac{c}{\sqrt{n \times n'}}. \quad (12.1)$$

The pair of structures that gives the smallest score is chosen to register  $g$  and  $g'$ . The resulting registered graphs are denoted  $g_a$  and  $g'_a$ .

### 12.3.1.2 Other Approaches to Registration of BGs

Deng et al. [12] in 2010, working with retina BGs, used a two-stage process for registration, also based on edge-to-edge correspondence. Their first (global) registration stage is also a RANSAC variant, where a vertex plus its neighbours in  $g$  is compared in  $g'$ . In practice, they restrict to degree 2 and 3 vertices, which corresponds to us choosing 2-stars and claws, respectively, as the structure (Their second stage registers vessel shape so is not in the scope of BGR). Using the BG vertex set, they compare the registration performance of several spatial topological graph structures commonly used in computer vision and graph-matching research: the Delaunay triangulation graph (DT), the minimum spanning tree of the DT graph, the  $k$ -nearest neighbour graph (KNN) and the minimum spanning tree of the KNN graph. They show that the BG technique substantially outperforms these other topological graph structures in graph registration, and state this is because BG characterises anatomical properties of the retinal vessels while the others do not.

Lupascu et al. [22], working with manually extracted retina BGs and  $S = E$ , enlarge the feature vector describing each edge from 2 to 9 dimensions by adding further spatial information relating to end vertices and midpoint of the edge, and vary  $f$  to be the Euclidean distance in 9-dimensional space. They set  $L = 30$  to test  $g$  against  $g'$  and also test  $g'$  against  $g$ , choosing only the edge pairs which appear in both lists. Then they use a quadric model to estimate the global transformation between the images using the endpoints of the matched edges.

Nibbelke [23], works with the earlier version of BGR ( $E, f, L, \varepsilon$ ) for finger vessel BGs. He systematically tests alternatives to steps 2 and 3 of the algorithm. First, he tries to improve the rough pre-orientation of images provided by the capture system by testing if the midline of the finger provides an intrinsic reference frame, but finds this not to be robust, leading to worse recognition performance than BGR in several experiments. Orienting all edges in the same direction before comparison does improve performance, as does sorting edge pairs using only their 1-dimensional difference in slope (i.e. using  $f = \Delta\theta$  and ignoring their difference in length). He also varies  $f$  to include weighting the difference in slope, to overcome the same problem of not finding the best edges for registration in the top  $L$ . His best results are found for  $f = \Delta\theta$ .

If an intrinsic reference frame does exist for pre-alignment in a particular vascular modality, it can be used to register the BGs. We have used this approach effectively with retina BGs in [5] (see Sect. 12.5) taking the centre of the optic disc as the centre of the graph coordinate system while the frame orientation is kept the same.

If no intrinsic reference frame exists for pre-alignment in a particular vascular modality, and we cannot even assume rough pre-alignment by virtue of the capture mechanism, then the BG may provide topological information we can use instead. We investigate this approach in our search for “anchors” in Sect. 12.5.

### 12.3.2 BGC-Biometric Graph Comparison

The second part of our algorithm is *noisy graph comparison*, to quantify the similarity between a pair  $g_a$  and  $g'_a$  of registered BGs. If we take advantage of the topology of the BGs in both the registration and noisy graph comparison algorithms, the speed and accuracy of graph comparison can be greatly enhanced.

The algorithm we use is based on using edges as structures as in [20], which is generalised in [6], and further generalised here. The BGC algorithm is flexible, so that any structure can be used. It has three steps: determination of the *minimum graph edit path* between  $g_a$  and  $g'_a$ , construction of the *Maximum Common Subgraph* (MCS) of  $g_a$  and  $g'_a$ , and finally, measurement of the difference between  $g_a$  and  $g'_a$  using the MCS.

We have previously demonstrated that the topology of MCSs generated from pairs of graphs from the same biometric instance (mated comparison) is different from that of MCSs generated from graphs from different instances (non-mated comparison) [6, 21].

The four design parameters in the BGC algorithm are: a *structure*  $S$ , cost matrix *weights*  $\alpha_1$  and  $\alpha_2$  used in the edit distance computation and *measure*  $d$  for scoring the distinctiveness or difference of  $g_a$  and  $g'_a$ . The structures  $S$  we have used are *Vertices* ( $V$ ), *Edges* ( $E$ ), *Claws* ( $C$ ) and *Two-claws* ( $T$ ). If we need to specify the parameters, we denote the algorithm by  $\text{BGC}(S, \alpha_1, \alpha_2, d)$ .

#### 12.3.2.1 BGC Algorithm Outline

**Step 1: Graph Edit Distance** The comparison process assumes that we have identified and listed all the structures of the selected type  $S$  in each registered graph. The registered graphs are compared using an inexact graph matching technique that computes the minimum cost *graph edit path* that converts  $g_a$  to  $g'_a$ . To do this, we use the Hungarian algorithm based method proposed by Riesen and Bunke [26]. One graph can be converted to another by 3 types of edit operations—insertions, deletions and substitutions. Each edit operation will incur a cost and the *graph edit distance* is the sum of the edit costs.

Selection of the right costs for these operations is critical to getting a meaningful measure of edit distance. The form of cost matrix we use is

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \\ \mathbf{C}_3 & \mathbf{C}_4 \end{bmatrix} \quad (12.2)$$

and depends on the choice of  $S$ . If the number of structures in  $g_a$  is  $m$  and in  $g'_a$  is  $m'$ ,  $\mathbf{C}$  is a  $(m + m') \times (m' + m)$  square matrix,  $\mathbf{C}_1 = [c_{ij} | 1 \leq i \leq m, 1 \leq j \leq m']$  and  $c_{ij}$  represents the cost of substituting structure  $u_i$  of  $g_a$  with structure  $v_j$  of  $g'_a$ . The sub-matrices  $\mathbf{C}_2$  and  $\mathbf{C}_3$  are square  $m \times m$  and  $m' \times m'$  matrices, respectively, with all elements outside the main diagonal equal to  $\infty$ . The diagonal

elements,  $c_{i\delta}$  of  $\mathbf{C}_2$  and  $c_{\delta j}$  of  $\mathbf{C}_3$  indicate the cost of deleting structure  $i$  from  $g_a$  and inserting structure  $j$  into  $g'_a$ , respectively.  $\mathbf{C}_4$  is an all zero matrix.

Cost matrix  $\mathbf{C}$  is fed into the suboptimal optimisation algorithm, which finds a local minimum edit cost. Output will be this lowest cost of converting  $g_a$  to  $g'_a$  and the list of edit operations that achieve it. The larger the number of structures in each pair of graphs, the bigger the matrices will be and the longer it will take the Hungarian algorithm to compute the optimum result.

The cost matrix entries we use depend on structure  $S$  and two weights  $\alpha_1$  and  $\alpha_2$ . The case  $S = V$  appears below as Example 12.1. Cost matrices for other structures are defined on similar lines (see Appendix 3), where  $\alpha_2$  will be weighted by the sum of the degrees of all the vertices in the structures.

*Example 12.1* (Vertex-based cost matrix, i.e.  $m = |V|, m' = |V'|$ .) Denote the degree of a vertex by  $D(\cdot)$  and the Euclidean distance between two vertex labels (spatial coordinates) by  $\|\cdot\|$ . The cost of substituting a vertex  $v_i$  of  $g_a$  with a vertex  $v'_j$  of  $g'_a$  is given by

$$c_{ij} = \|v_i, v'_j\| + \varpi_{ij}. \tag{12.3}$$

where  $\varpi_{ij}$  is the cheapest cost obtained as output when applying the Hungarian algorithm on a cost matrix for subgraphs  $g_{v_i}$  and  $g'_{v'_j}$  (see [7] for details). These subgraphs are constructed from the vertices  $v_i$  and  $v'_j$  and their first-hop neighbourhoods, respectively. The total cost of deleting a vertex will be the sum of the cost of deleting the vertex itself ( $\alpha_1$ ) and the cost of deleting its neighbourhood vertices ( $\alpha_2$  for each neighbouring vertex),

$$c_{i\delta} = \alpha_1 + (\alpha_2 \times D(v_i)). \tag{12.4}$$

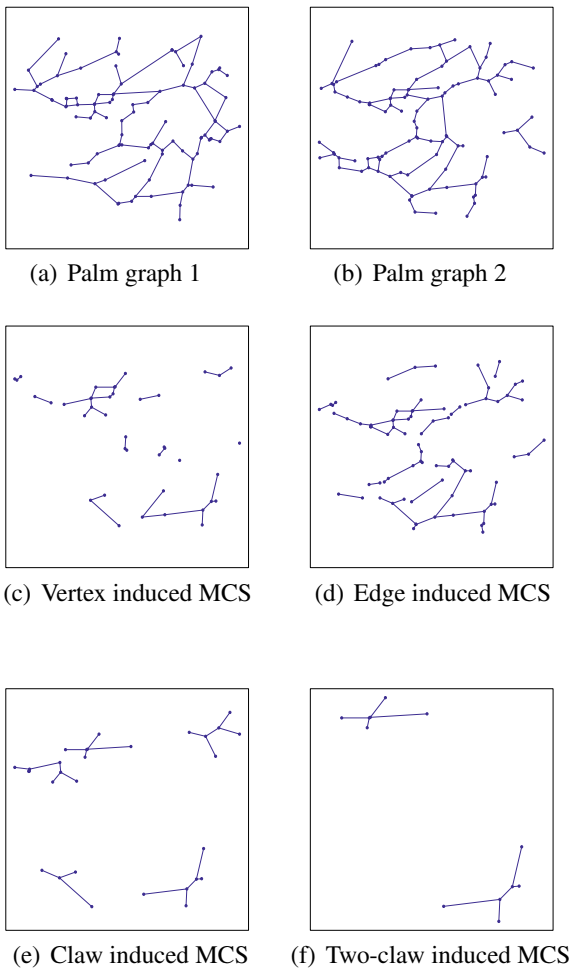
Similarly, the cost of inserting a vertex is

$$c_{\delta j} = \alpha_1 + (\alpha_2 \times D(v'_j)). \tag{12.5}$$

**Step 2: Maximum Common Subgraph** We use the locally optimal edit path output by the Hungarian algorithm to define a subgraph of  $g'_a$ . It includes all those structures of  $g'_a$  that are included in the list of substitutions. The structures deleted from  $g_a$  and the structures inserted into  $g'_a$  are excluded, but any additional corresponding edges are included. This subgraph is called the Maximum Common Subgraph (MCS) of  $g_a$  and  $g'_a$  as it represents all those structures in  $g'_a$  that are “matched” to structures in  $g_a$ . We also call it an *S-induced subgraph* of  $g'_a$  as the subgraph is induced by the substituted structures in  $g'_a$  (Note that defining the MCS as a subgraph of  $g_a$  is equivalent.).

**Definition 12.2** Assume  $BGC(S, \alpha_1, \alpha_2, -)$  has been applied to registered graphs  $g_a$  and  $g'_a$  in Step 1 above. Their ( $S$ -induced) *Maximum Common Subgraph (MCS)* is the subgraph of  $g'_a$  consisting of all structures in  $g'_a$  that are included in the list of

**Fig. 12.2** This figure shows the Maximum Common Subgraph between the palm vessel graphs in **a** and **b** resulting from applying BGC with the structure  $S$  to be **c** vertices, **d** edges, **e** claws and **f** two-claws. Vertex- and edge-induced MCSs are bigger than claw- and two-claw-induced MCSs as the conditions for the structures to match in the former cases are not as strict as in the latter



substitutions, *together with* any edges that exist between these substituted structures in  $g'_a$ , for which a corresponding edge exists in  $g_a$ .

Depending on the structure used, the MCS can be vertex induced, edge induced, claw induced or two-claw induced. Figure 12.2 shows each type of MCS for a typical pair of palm BGs from the same biometric instance. The edge induced MCS is the most connected with the richest structure of the four. As  $S$  gets more complex than  $E$ , the corresponding MCS will be sparser, but the nodes and edges that form part of the MCS will be more reliable. In our experience, the node-induced subgraph tends to miss out on some of the structure that is present in the edge-induced subgraph. Therefore, overall for the biometric graphs in the databases we studied, we prefer  $S$  to be edges.

**Table 12.1** Difference measures between  $g_1$  and  $g_2$ , determined by counts of structures in their MCS

$d$	$d_v$	$d_e$	$d_t$	$d_{c_1c_2}$	$\rho_{c_1}$
$M$	$ V_m $	$ E_m $	$ T_m $	$ V_{c_1}  +  V_{c_2} $	$ E_{c_1} $
$N_i$ or $N$	$ V_i $	$ E_i $	$ T_i $	$ V_i $	$ V_{c_1} $

**Step 3: Difference Measures** The MCS topology is used to define difference measures between  $g_a$  and  $g'_a$ . There are many potential score functions to separate genuine and impostor comparisons. We have tested 20 which are described in Sect. 12.4.3. A selection of 5, that have proved the most effective, is presented in Table 12.1. One of them, the Bunke–Shearer metric  $d_v$ , is already known [9]. Call the two aligned graphs being compared  $g_1 = (V_1, E_1)$  and  $g_2 = (V_2, E_2)$ , with  $g_m = (V_m, E_m)$  as their MCS. All sets from  $g_i$ ,  $i \in \{1, 2, m\}$ , are subscripted with  $i$ . Corresponding sets used to define the measures are the vertex set  $V_i$ , the edge set  $E_i$  and the set of two-claws  $T_i$ . We are also interested in  $c_i = (V_{c_i}, E_{c_i})$ ,  $i = 1, 2$ , the first and second largest connected components of  $g_m$ . The measures have two forms, a distance

$$d = 1 - \frac{M}{\sqrt{N_1 \times N_2}} \tag{12.6}$$

or density

$$\rho = M/N \tag{12.7}$$

as detailed in Table 12.1.

The previous Sections have dealt with the formal aspects of vascular Biometric Graph Comparison. In the next Section, we summarise the performance and practical advantages and disadvantages already discovered using BGC.

## 12.4 Results

This section will describe the public vascular databases used for BGC so far and compare key BG statistics across them. We summarise experimental results we have obtained by applying BGC to BGs from databases of the four modalities we have studied. The important outcomes from this work are

- that using graph structure in the registration algorithm can increase the speed and accuracy of registration;
- that graph structure in the MCS can be exploited to increase recognition accuracy; and
- that using multiple graph structures can improve similarity scores over single structures.

### 12.4.1 *Vascular Databases*

To our knowledge, the BGC algorithm has been tested on five vascular modalities: *Palm* vessels representing the vascular pattern under the palm of the hand; *Wrist* vessels representing the vascular pattern on the inside of the wrists; *Hand* vessels representing the vascular pattern under the skin on the back (dorsal surface) of the hand; *Retina* vessels representing the vascular pattern supplying blood to the retina; and *Finger* vessels representing the vascular pattern under the skin of the finger. We have tested the first four modalities. Finger vessel has been tested by Nibbelke [23], who found that in this case BGC was not competitive with standard point pattern comparison techniques. Goulu [16] in his work on *Face* vessels representing the vascular pattern under the skin of the face, uses a database collected by the University of Houston and extracts BGs. He claims to test BGC but no details are given in [16].

Details of the databases used are summarised in Table 12.2. All are either available for download or on request from the researchers who collected them. The palm and wrist image databases are obtainable from the Poznan University of Technology (PUT) [18] and can be downloaded at <http://biometrics.put.poznan.pl/vein-dataset>. The hand image databases are from Singapore's Nanyang Technical University [27] with images captured in the near-infrared (SNIR) and far-infrared (SFIR) wavelengths over three sessions each separated by a week. This database exemplifies the kind of variation that can be expected in captures taken across sessions. This is typical of a biometric scenario, where translation and rotation of the images occur between captures due to human factors. Access to this database was obtained by emailing the authors of [27]. Retina images are from the publicly available VARIA database [24] accessible at <http://www.varpa.es/research/biometrics.html>. In Sect. 12.5 we also refer to the ESRID retina database collected by RMIT University (c.f. [2]). This database can be accessed by emailing the second author of [2]. The finger image database used by Nibbelke [23] is from the University of Twente (UT) and can be accessed by emailing the lead author of [23].

### 12.4.2 *Comparison of Graph Topology Across Databases*

In principle, there is no restriction on the structure used by the BG registration and comparison algorithms. In practice, there are restrictions imposed by both the physical form of the vasculature and by the limitations of image resolution and image processing. How do we know what range of options is available?

We have already noted the visible similarity of vascular graphs to trees or ladders. This results from the way the vasculature forms physically. Its purpose is to deliver blood to and from tissues, with the capillaries forming the very fine vessels connecting the arterial system to the venous system. Capillaries are so fine that this interconnection is lost in many images, and vessels appear to terminate rather than

**Table 12.2** Vessel image databases used for BGC

Database	Subjects $\times$ instances	No. of sessions	Samples/session	Total samples
PUT palm	50 $\times$ 2 (left, right)	3	4	600
PUT wrist	50 $\times$ 2 (left, right)	3	4	600
SFIR hand <sup>a</sup>	34	1	$\geq 2$	173
SNIR hand <sup>a</sup>	123	1	$\geq 2$	732
VARIA retina <sup>a</sup>	37	1	$\geq 2$	97
ESRID retina	46	1	9	414
UT finger	60 $\times$ 6 (different fingers)	2	2	1440

<sup>a</sup>Subset obtained after removal of subjects who had only 1 sample present

rejoin. Typically, vessels do not branch into more than two sub-branches at the same point. As well, while distinct principal veins and arteries might enter the biometric ROI at separate points, all of the vasculature derived from each such vessel will be connected. No sub-branches will actually be disconnected from a parent vessel.

Consequently, in a BG that is perfectly extracted from a high-quality two-dimensional vascular image, there will be relatively few cycles, which will mostly result from vessel crossovers. Vertices will have a low degree (most likely  $\leq 4$  with maximum degree 4 occurring at crossovers). There will be no isolated vertices (i.e. minimum degree will be 1) and the ratio of edges to vertices (the *density* of the BG) will be similar to that of a tree and so, close to 1. The BG will be connected.

In practice, the image quality will affect the connectivity of the BG, as the image processing algorithm will be unable to extract features from poor quality regions of the image. The more complex the structure of interest, the greater the chance that an occurrence of it will not be extracted in the BG from a particular image, because a component vertex or edge is missing as a result of noise in the image, or suboptimal sensing or image processing. For this reason we are also interested in the largest connected component  $C_1$  of the BG. The size of the largest component is an indication of the amount of noise in the image that has not been compensated for by the image processing.

### 12.4.2.1 BG Statistics

A very basic question is how much the underlying BG statistics vary for different databases for the same modality, as well as how much they vary for different modalities. In Table 12.3, we record fundamental statistics for different BG databases: numbers of vertices, edges, claws and two-claws, density and number of vertices in the largest connected component  $C_1$  of the BG.



**Table 12.3** Mean (standard deviation) of BG topologies for each database. All data except for the last row appear in [6]. Here  $V$  is the vertex set,  $E$  the edge set,  $C$  the claw set,  $VC_1$  the vertex set of the largest connected component  $C_1$ , and  $T$  the two-claw set

Database	$ V $	$ E $	$ E / V $	$ VC_1 $	$ C $	$ T $
Palm Left	103 (27.9)	109.9 (19.47)	1.05 (0.03)	93.7 (21.68)	37.6 (7.97)	28.21 (8.39)
Palm Right	98.5 (18.8)	104.1 (20.75)	1.05 (0.03)	89.5 (21.35)	35.4 (8.3)	26.39 (8.53)
Wrist Left	83.07 (16.66)	86.2 (18.61)	1.03 (0.04)	72.1 (19.76)	27.01 (7.85)	19.18 (8.14)
Wrist Right	81.4 (16.11)	83.9 (17.82)	1.02 (0.04)	70.5 (19.43)	25.6 (7.31)	17.66 (7.54)
Hand SFIR	51.7 (9.38)	48.8 (9.6)	0.94 (0.06)	34.6 (12.94)	21.6 (5.35)	19.94 (6.59)
Hand SNIR	39.4 (13.07)	37.6 (13.1)	0.95 (0.03)	27.9 (11.76)	17.3 (6.38)	15.36 (6.47)
Ret. VARIA	70.3 (27.9)	67.1 (29.23)	0.94 (0.07)	48.6 (20.98)	28.9 (13.7)	29.65 (15.28)
Ret. ESRID	146.2 (86.7)	152.6 (92.7)	1.03 (0.04)	109 (68.9)	73.3 (45.1)	75.4 (48.7)

Table 12.3 shows some interesting differences and similarities between the different vascular graphs. All the graphs have density quite close to 1, reflecting their similarity to trees, as expected. The maximum degree of a vertex for each BG was also determined but not recorded here as for every database the mode of the maximum degrees is 3. Between 30 and 40% of vertices in the BGs on average in every database form claws. This indicates that bifurcations are commonplace in our vascular modalities while crossovers are not as commonly seen.

Within modalities, the far-infrared images (SFIR) for hand vessels are superior to the near-infrared (SNIR) as far as being able to extract BGs with usable structure is concerned. With retina, the ESRID graphs are much larger and more connected than VARIA graphs. There is also a large variation across the sizes of the graphs in ESRID when compared to VARIA. The probability of finding a two-claw structure in a retina BG is higher on average than for the other modalities.

The hand BGs are, nonetheless, the smallest and least structured of all modalities, with lower connectivity evidenced by only 70% of their vertices belonging to the largest component. The palm BGs are the second largest (after retina BGs) and most structured, with a higher connectivity than the other graphs demonstrated both by density and the fact that over 90% of the vertices belong to the largest component.

#### 12.4.2.2 Proximity Graphs

Another topological measure we use to characterise the different BG modalities is the distance a BG is from a *proximity graph* on the same vertex set. Proximity graphs were defined by Davis et al. [11]. A proximity graph  $p_\varepsilon$  on spatial vertex set  $V$  is one where a pair of vertices in  $V$  have an edge between them if and only if they are less than  $\varepsilon$  units apart. That is, for a proximity graph, the edges are completely defined by the spatial arrangement of its vertices. The closer a graph is to a proximity graph, the more predictable its edges are.

**Table 12.4** [6] The mean (standard deviation) distance of a BG to its nearest proximity graph

Database	Proximity graph distance
Palm	0.017 (0.004)
Wrist	0.022 (0.006)
Hand SFIR	0.032 (0.007)
Hand SNIR	0.046 (0.018)
Retina VARIA	0.031 (0.010)

Thus, if  $g = (V, E, \mu, \nu, A)$  is a BG there is a family of proximity graphs  $\{p_\varepsilon, \varepsilon \geq 0\}$  defined by  $V$ . For each  $\varepsilon$ , a normalised distance between  $g$  and  $p_\varepsilon$  can be determined from their adjacency matrices, using formulas described in [11]. The value of the proximity graph distance varies from 0 to 1, where zero implies that the graph is a proximity graph. The minimum of these distances over the available range of  $\varepsilon$  decides the specific value of the bound  $\varepsilon$  and the closest proximity graph  $p_\varepsilon$  to  $g$ . Table 12.4 shows the average and standard deviation of this distance from a BG to its nearest proximity graph, for each of the databases.

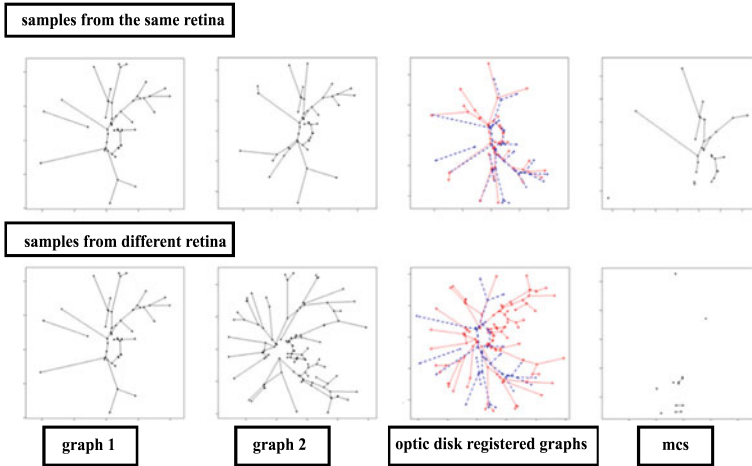
The BGs from palm and wrist vessels have the lowest average distances to a proximity graph, implying that their edges are more predictable than the other BG modalities. Edges are more likely to occur between nearby vertices in palm and wrist BGs than for other modalities, which suggests that the relational information in the graph representation is less surprising (has lower entropy). In principle, the higher the distance, the more promising the vascular pattern is as a biometric modality.

### 12.4.3 Comparison of MCS Topology in BGC

In previous work [6–8, 20, 21], we have investigated many potential structures and graph statistics in MCSs for their usefulness in BGC for finding information that will satisfactorily separate genuine MCSs from impostor MCSs. Genuine MCSs usually look quite different from impostor MCSs, the latter appearing fragmented and broken as seen in Fig. 12.3. We have attempted in numerous ways to find measures that capture this visually striking difference.

Here, we summarise our findings and discuss reasons for restricting to the structures and corresponding similarity score measures we now use.

Our initial application of BGC [7] was to the retina modality, which has been repeatedly shown (on very small databases) to have high accuracy, with complete separation of genuine and impostor scores typically being demonstrated for vertex comparison approaches. In [7], with manually extracted BGs from the VARIA retina database, we introduced the original BGC (with  $S = V$  in the comparison step). We tested 8 measures based on the MCSs for both genuine and impostor comparisons. The 6 normalised quantities were  $d_v$ ,  $d_e$  and the differences  $n_2$ ,  $n_3$ ,  $p_2$ ,  $p_3$  using Eq. (12.6) corresponding to numbers of vertices of degree  $\geq 2$ , vertices of degree



**Fig. 12.3** This is an example of the BGC algorithm when two samples from the same retina instance are compared (genuine comparison) versus when two samples from different retina instances are compared (impostor comparison). Note the MCSs are visually different, with the genuine MCS having more vertices and a more complex structure than the impostor MCS

$\geq 3$ , paths of length 2 and paths of length 3 in  $g_1$ ,  $g_2$  and  $g_m$ , respectively. The 2 unnormalised quantities were the density  $\rho_m = |E_m|/|V_m|$  of  $g_m$  and the variance  $\sigma_D^2$  of the degree distribution of  $g_m$ . Of these, the score distances for genuine comparisons using vertices of degree  $\geq 3$  and paths of length 3 were too high to warrant further use. Vertices of degree  $\geq 2$  and paths of length 2 were also not further considered, as they correlated too highly with either  $d_v$  or  $d_e$ .

Score fusion using  $d_v$  and  $d_e$  gave better, but not significantly better, performance than either single measure, probably because these measures are highly correlated. In fact the least correlated measures are  $d_v$ ,  $\rho_m$  and  $\sigma_D^2$ . These measures completely separated scores in two or three dimensions, an improvement on separation in one dimension which is expected to become significant in larger retina databases.

In [20], we developed the first full BGC system to automatically extract retina BGs and compare them, again using the VARIA database. Our intention was to see if the results of [7] could be improved using automatic extraction of BGs. We retained the measure  $d_v$ , introduced  $d_{c_1c_2}$  based on the two largest connected components of  $g_m$ , and replaced  $\sigma_D^2$  by the maximum degree  $D_{max}$  of a vertex in  $g_m$  (another unnormalised quantity). Again we showed that using  $d_v$  alone gave complete separation in the training set. Using two or all three measures in a combination of an SVM classifier and KDE curves [20] or surfaces gave dramatic improvements in False Match rate (FMR) (up to several orders of magnitude), when False Non-Match Rate (FNMR) was very low.

For hand vessel BGs using the SNIR and SFIR databases in [21], we tested the 7 measures  $d_v$ ,  $d_e$ ,  $|V_{c_1}|$ ,  $|V_{c_1}| + |V_{c_2}|$ ,  $\sigma_D^2$ ,  $D_{max}$  and, for the first time, the average degree  $\mu_D$  of the vertices in the MCS. The best-separating individual measures

were  $d_v$ ,  $d_e$  and  $|V_{c_1}| + |V_{c_2}|$ , but as  $d_v$  and  $d_e$  are highly correlated, the relatively uncorrelated measures  $d_v$ ,  $|V_{c_1}| + |V_{c_2}|$  and  $\sigma_D^2$  were tested to see if multidimensional scoring would improve performance over individual measures. In contrast to the case for retina, we found little advantage in increasing the number of measures used. We attribute this to the fact that hand BGs are appreciably smaller and more fragmented than retina BGs (see Table 12.3 and [21, Fig. 3]) and will have correspondingly less available topology in their MCSs to exploit.

As a consequence of these experiments, the measures we focussed on were  $d_v$ ,  $d_e$ ,  $d_{c_1c_2}$ ,  $\rho_m$ ; and  $d_{c_1}$  and  $d_{c_2}$ , the measures using Eq. (12.6) corresponding to the number of vertices in  $c_1$  and  $c_2$ , respectively.

For the larger palm vessel BGs, in [8] we test these 6 measures<sup>3</sup> and a further 4:  $\rho_{c_1}$ ; the ratio of the number of isolated vertices  $I$  to the number of connected vertices; the normalised total length  $d_\ell$  of the edges in  $c_1$ ; and the ratio  $n_4$  of the number of vertices with degree  $\geq 4$  in  $g_m$ , to  $|V_m|$ . Equal Error rates using single measures were competitive (under 5%) for within session comparisons for the measures  $d_v$ ,  $d_e$ ,  $d_{c_1}$ ,  $d_{c_1c_2}$ ,  $\rho$  and  $d_\ell$ , with three of these,  $d_v$ ,  $d_e$  and  $d_{c_1c_2}$ , having competitive EERs across sessions as well. The measure  $d_e$  outperformed all others. Testing score pairs showed that pairing  $d_e$  with any of  $d_{c_1}$ ,  $d_{c_1c_2}$  and  $d_\ell$  improved performance over the single score  $d_e$ , with  $(d_e, d_\ell)$  having the maximum gain.

In [6], we tested our ideas on all four modalities using a uniform approach. Our results are outlined in the Sect. 12.4.4, which explains the selection of difference measures in Table 12.1.

- Our attempts to quantify our observation that higher degree vertices occur more frequently in genuine MCSs than in impostor MCSs ( $n_2, n_3, \mu_D, \sigma_D^2, n_4$ ) coalesced in the single measure  $d_C$  of claws (i.e. of degree 3 vertices).
- Our efforts to quantify our observation that connected components are larger in genuine MCSs than in impostor MCSs led to the measures  $d_{c_1}$ ,  $d_{c_2}$ ,  $d_{c_1c_2}$ ,  $d_I$ .
- Our wish to capture some spatial information rather than counts alone resulted in  $d_\ell$  and a new measure  $d_a$  found using Eq. (12.6) from the area of the smallest rectangle containing the entire graph.
- Our efforts to quantify our observation that genuine MCSs have higher complexity than impostor MCSs led us to use  $\rho_m$ ,  $\rho_{c_1}$ ,  $D_{max}$  and a new measure  $d_t$  using Eq. (12.6) for the number of two-claws.

For convenience this subsection is summarised in Table 12.5. Measures that we have only tested once before 2017 ( $p_2, p_3, \mu_D, n_4$ ) are not included. Plainly this topic is by no means exhausted.

### 12.4.4 Comparison of BGC Performance Across Databases

In this subsection, we outline the results and conclusions of our paper [6], in which we evaluated the performance of BGC for the databases of Sect. 12.4.1. The individuals

<sup>3</sup>In fact the corresponding similarity measure  $1 - d$  was used for the normalised measures.

**Table 12.5** Difference measures used in BGC

Year	Mode	$d_v$	$d_e$	$d_C$	$d_{c_1 c_2}$	$d_{c_1}$	$d_{c_2}$	$\rho_m$	$\rho_{c_1}$	$\sigma_D^2$	$D_{max}$	$d_I$	$d_\ell$	$d_a$	$d_t$
2011 [7]	Retina	✓	✓	✓ <sup>a</sup>				✓		✓					
2013 [20]	Retina	✓			✓						✓				
2014 [21]	Hand	✓	✓		✓ <sup>b</sup>	✓ <sup>b</sup>				✓	✓				
2015 [8]	Palm	✓	✓		✓	✓	✓	✓	✓			✓ <sup>c</sup>	✓		
2017 [6]	All <sup>d</sup>	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓

<sup>a</sup> $n_2$  and  $n_3$  count the degree  $\geq 2$  and  $\geq 3$  vertices.  $d_C$  counts the degree 3 vertices (or claws)

<sup>b</sup>Un-normalised counts used

<sup>c</sup>Ratio of  $I$  to number of connected vertices, rather than normalised using Eq. (12.6)

<sup>d</sup>Retina, hand, palm, wrist

in each of the five databases (2 for hand) were divided in two, with BGs for one half used for training and the other for testing, to maintain independence. For full details of the experiments, see [6].

The first training experiment was to tune for BGR: to identify the best structure  $S \in \{E, C, T\}$  for graph registration for each database, the optimal pair shortlist length  $L$  and the tolerance  $\epsilon$ . This list was selected based on observation. For each  $S, L$  was varied by steps of 40 through the range [20, 220]. Because accurate registration is crucial to the performance of BGC, we selected the  $L$  leading to highest registration accuracy. There is a consequent trade-off in speed versus accuracy, as Table 12.6 demonstrates.

The second training experiment was to tune the parameters of BGC: the structure  $S \in \{V, E, C, T\}$  and parameters  $\alpha_1, \alpha_2$  for the graph edit computations and the difference measure  $d$  for scoring MCSs. The parameters were each stepped by 2 in the range [3, 9]. For each database, a subset of 1000 genuine and 1000 impostor comparisons was selected at random and their MCSs computed and scored with the 13 graph measures (see Table 12.5) to find the values giving optimal separation. To check if any combination of measures would improve separation, we combined all 13 measures and used LDA to check this, but found no significant improvement over single measures. For all databases, selecting  $V$  for the cost matrix structure and  $d_v$

**Table 12.6** [6] The chosen registration structures  $S$  and shortlist values  $L$  for each database and the average registration times

Database	$S$	$L$	Time (s)
Palm	$E$	220	20
Wrist	$E$	60	7
Hand SFIR	$T$	60	0.8
Hand SNIR	$E$	60	1.9
Retina VARIA	$T$	100	1.8

**Table 12.7** [6] The graph matching parameters chosen based on best performance on the training set

Database	$\alpha_1$	$\alpha_2$	Best $d$	2nd best $d$	3rd best $d$
Palm	5	3	$d_e$	$d_v$	$\rho_{c_1}$
Wrist	3	7	$d_e$	$d_v$	$\rho_{c_1}$
SFIR Hand	3	7	$d_v$	$d_e$	$d_t$
SNIR Hand	3	5	$d_v$	$d_e$	$\rho_{c_1}$
Retina VARIA	3	9	$d_v$	$d_e$	$d_{c_1c_2}$

**Table 12.8** [6] Comparison performance using BGC on the test set at 2 specific thresholds obtained from the *training set experiments*—FMR100 and FMR1000

Threshold	Palm	Wrist	Hand—SFIR	Hand—SNIR	Retina
	FNMR %	FNMR %	FNMR %	FNMR %	FNMR %
FMR100	3.63	26.9	4.39	0.54	0.07
FMR1000	6.242	44.06	8.79	99.72	0.86

or  $d_e$  gave the best separation. Table 12.7 summarises the results. The five graph measures on the MCS that we found to be the best difference measures, are  $d_v$ ,  $d_e$ ,  $\rho_{c_1}$ ,  $d_{c_1c_2}$  and  $d_t$ .

After tuning, we tested BGC on the remaining half of the individuals and determined FMR and FNMR of comparisons at three distance thresholds chosen from the training experiments—EER, FMR100 and FMR1000. ROCs for the SNIR Hand vein database training set do not appear in [6] and are given in Appendix 4. All databases other than the wrist, gave error rates under 5% at the EER threshold. Those for palm, hand and retina were comparable with our previous results or the literature. Table 12.8 records our results.

We have already shown for hand vessels [21] that including edge information in BGC improves recognition performance over point pattern comparison. Our final experiment was to apply ICP to register graph pairs, then apply Step 4 of BGR to count matched vertices in the two graphs, again scoring using QuickScore (Eq. (12.1)) for consistency. In all cases, BGC outperformed point pattern comparison using ICP registration. See Table 6 of [6] for exact values.

## 12.5 Anchors for a BGC Approach to Template Protection

The purpose of biometric authentication is to link a subject unequivocally to the authentication token. The biometric template used to form the token comprises personal and sensitive information and is often encrypted when stored. However, as biometric data is noisy, comparison with an incoming biometric sample cannot be done in the encrypted domain using cryptographic hash functions as these require exactness of data. Consequently, most authentication systems decrypt the stored bio-

metric data, compare the unencrypted templates and make an authentication decision. This makes the biometric template vulnerable during comparison.

Thus, finding a template protection scheme which permits direct comparison of protected templates is desirable. In any such scheme, performance degradation over unprotected comparison must be marginal. Further, the ISO/IEC 24745:2011 standard [25] states the following two criteria to protect biometric information: (a) *Irreversibility* where the biometric raw data cannot be retrieved from the template or token, and (b) *Unlinkability* where multiple independent instances of a subject cannot be linked to identify the subject.

We are interested in the possibility of using biometric graphs in a template protection scheme based on a dissimilarity vector model.

### 12.5.1 Dissimilarity Vector Templates for Biometric Graphs

We want to investigate the feasibility of protecting a BG template by representing it as a vector of dissimilarities from a fixed set of reference BGs extracted from a separate, external set of instances. Such reference graphs are termed “cohorts”. The reason that cohort-based dissimilarity vectors may be a solution to having template-protected biometric comparison for automatic identity authentication is that the biometric sample data need not be stored. Only the cohort graphs and the dissimilarity vector are required for authentication. On the face of it, neither of these reveal any direct information about the biometric sample data of enrolled individuals.

In preliminary work [5], we use retina as an example to conduct the first step of this investigation: to test if the comparison performance of the dissimilarity vector templates is similar to that of unprotected template comparison.

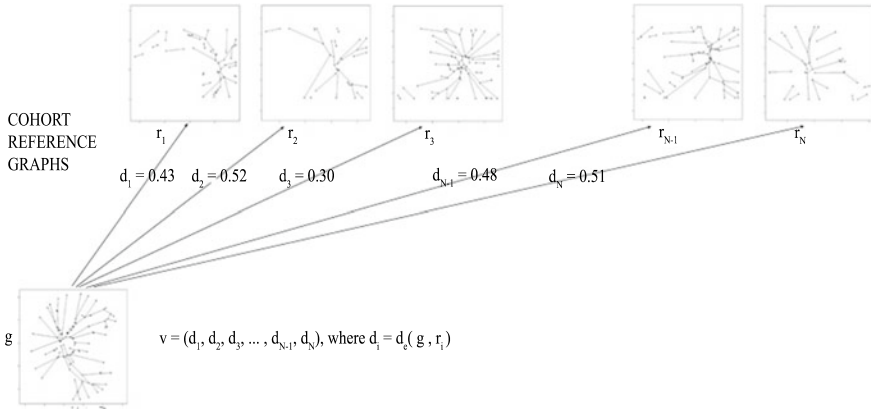
Cohorts are typically not used in existing dissimilarity vector implementations because of the expectation that graphs which are not a member of any class will be dissimilar to all classes and hence not useful for classification. Contrary to this, we found that when retina graphs are registered on the optic disc then graphs extracted from images of the same retina are surprisingly and consistently dissimilar, or similar, to other retina graphs external to the classification set, when the dissimilarity is defined by the BGC algorithm with slack graph comparison parameters.

Figure 12.4 shows an example of a dissimilarity vector for a retina graph.

We have shown that the dissimilarity vector approach is accurately able to compare and verify samples with only a small loss in performance over direct comparison using BGC. Once performance is established, the next step would be to establish rigorous security bounds on irreversibility and unlinkability as conducted by Gomez-Barrero et al. [14, 15]. This is an area of future work.

### 12.5.2 Anchors for Registration

Amongst the modalities presented here, retinae have an intrinsic reference frame defined by the location of optic disk and fovea. Palm vein patterns have a reference



**Fig. 12.4** An example of a dissimilarity vector for a retina graph  $g$  in ESRID from a set of cohort graphs in VARIA. The dissimilarity vector  $v = (d_1, d_2, \dots, d_N)$  is the vector of dissimilarities from the ordered set of cohort graphs  $(r_1, r_2, \dots, r_N)$ . Each  $d_i \forall 1 \leq i \leq N$  is calculated as  $d_i = d_e(g, r_i)$ , where  $d_e$  is some measure of dissimilarity between graphs  $g$  and  $r_i$

frame defined by the hand contour. For other vascular patterns, an intrinsic reference frame has not been identified (for finger graphs, the midline of the finger was found by Nibbelke [23] not to be robust), and because of the noise associated with presentation of a biometric sample and graph extraction, graphs extracted from images from the same individual do not consistently register with reference graphs in the same way when using BGR and are not consistently dissimilar. The retina graphs in both the ESRID and VARIA databases are roughly pre-aligned because the presentation of the retina is always with the head upright, and so a common reference frame for a pair of retina graphs extracted from these images can be found by centring each graph on the centre of the optic disk (also extracted from the associated retina image).

Hence, a barrier to generalising the dissimilarity vector approach to template protection to other vascular graphs is the ability to register presentations of a vascular pattern from the same individual in the same way so that their dissimilarity from a set of reference graphs has the possibility to be consistent. The alternative, which is to use BGR, gives a set of scores that are essentially drawn from a distribution of impostor comparison scores and are different from one sample to the next.

In an attempt to achieve consistent registration, we consider identifying subgraphs of a smaller size that are consistently extracted in multiple presentations of a subject’s biometric data despite the noise in the image presentation and extraction process. We term this small subgraph, should it exist, the *anchor* for a set of biometric graphs from an individual.

**Definition 12.3** A BG *anchor* for an individual is a small connected subgraph that appears consistently in BGs extracted from multiple good samples from the individual and that does not by itself reveal identifying information about the individual.



Whether such an anchor exists for every enrolled subject is the first question, which we attempt to answer here for two of the databases we have studied. Whether registration on such an anchor then leads to dissimilarity vectors that can be used for accurate classification is a separate question and is future work.

### 12.5.3 The Search for Anchors

The BGC algorithm can be used recursively to find anchors. Let  $g_1, g_2, \dots, g_n$  be the BGs of the  $n$  samples of a subject for which we need to find an anchor.

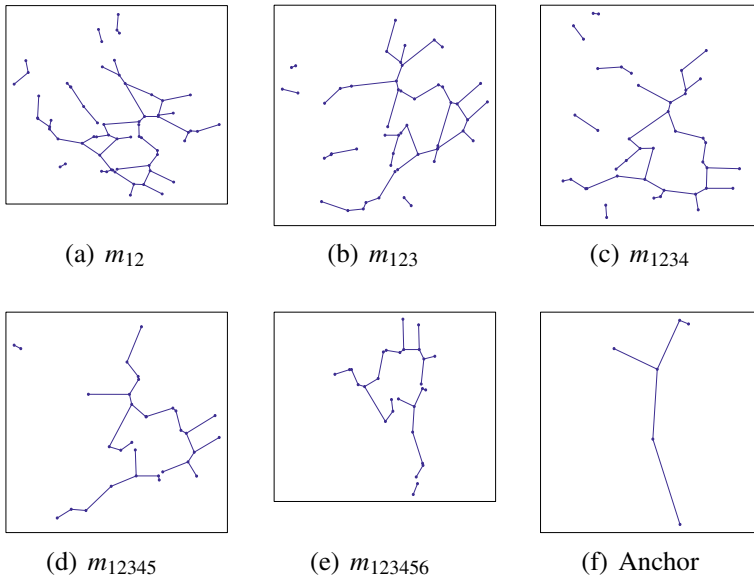
The first step is to use the BGC algorithm to find the MCS between a pair of graphs. Let  $m_{12}$  be the MCS of the graphs  $g_1$  and  $g_2$ . BGC is then used to find the MCS between  $m_{12}$  and the third graph in the list  $g_3$ . Let this be denoted by  $m_{123}$ . This is the common graph between  $g_1, g_2$  and  $g_3$ . If we continue this process, the common graph between the  $n$  graphs  $g_1, g_2, \dots, g_n$  is the MCS between  $m_{123\dots n-1}$  and  $g_n$  and is denoted by  $m_{123\dots n}$ . This graph represents the graph structure that is common to the  $n$  samples from a subject. If the graph samples are of high quality, we often find this common graph to be large with significant amount of structure. Therefore, the entire common graph would be inappropriate to use as an anchor associated with a template protection scheme. On the basis of observation and experimentation, we have isolated two criteria to derive an anchor from  $m_{123\dots n}$ :

- It is the largest connected component of  $m_{123\dots n}$  that has a minimum of at least 5 vertices and maximum of 10 vertices. This criteria ensures that the anchor is not so large as to reveal significant structure of a subject's BG.
- This connected component must have at least one claw. In cases where there was an absence of a claw (i.e. the component was a path) we observed that the anchor was not uniquely found.

One way to satisfy the above two criteria is to vary the weights  $\alpha_1$  and  $\alpha_2$  in the cost matrix  $\mathbf{C}$  of the BGC algorithm used when finding anchors. When  $\alpha_1$  and  $\alpha_2$  are small, the MCS returned will be very small and sparse. As we want to have recursively generated MCSes to have a bit more structure, we found it beneficial to recursively slacken  $\alpha_2$  until we find a common graph of the  $n$  graphs that will give an anchor that satisfies the above two conditions.

To study the possibility of finding anchors and the various factors that impact this for a database, we need a database that has multiple samples of the same subject. The PUT datasets of palm and wrist vessels had 12 samples per subject across 3 sessions and were satisfactory for our experiments.

For both databases we chose  $n$ , the number of graphs of a subject used to find an anchor, as  $n = 6$ . We used the remaining 6 samples as test samples to determine if an anchor can be found in a new incoming sample. We set  $\alpha_1 = 1$  in the cost matrix  $\mathbf{C}$  and recursively increased  $\alpha_2$  from 4 to 16 in steps of 2 in the anchor-finding algorithm.



**Fig. 12.5** This figure shows the common graphs and the final anchor obtained when BGC is used recursively, pairwise on a set of BGs from an individual in the PUT palm database to create the anchor for that individual. Observe that as expected, the size of the common graph as we increase the number of BGs gets smaller. **f** shows the extracted anchor (Graphs are not on the same scale)

Figure 12.5a–e shows the process of recursively applying the BGC algorithm to obtain a common graph among 6 BGs of a subject in the PUT Palm database. We observe that as the number of samples used increases, the common graph tends to get smaller and sparser compared to previous common graphs. For a graph to become part of the common graph it must exist in all the BGs used to form it. The criteria get harder to satisfy as the number of BGs increase. Figure 12.5f and shows the anchor, a subgraph of  $m_{123456}$  in Fig. 12.5e, which is the largest connected component of maximum order 10 with at least one claw.

### 12.5.4 Queries and Discoveries for Anchors

To understand if the use of anchors is practical for registering BGs, we used the palm and wrist databases to investigate the following questions:

1. How likely is it that an anchor cannot be found for a subject in the database and what are the possible reasons for failure to find an anchor?
2. If an anchor is generated using a few samples of a subject, how do we determine if it exists in a new probe sample of the same subject. How reliable is this anchor?

3. How often will an anchor fail to be found in a new probe sample of an enrolled subject? If this happens, what are the causes?

For both databases, we chose 6 BGs from the 12 BGs of each subject in 4 ways giving 4 different attempts at finding an anchor. As the PUT database had 50 subjects, we had 200 trials to find an anchor and we noted the number of trials that failed to find an anchor (first column of Table 12.9).

Once an anchor is found, it needs to be reliably found in a new sample of the same subject. The existence of an anchor in a larger graph can be determined using the BGR algorithm described in Sect. 12.3.1.1. The BGR algorithm will attempt to find an aligning edge between the anchor and a BG of an individual. *Anchor overlap* is defined as the fraction of vertices in the anchor that found a comparison vertex in the BG. 100% overlap indicates the anchor has been exactly found in the BG and can be reliably used to establish a coordinate frame of registration. Figure 12.6 shows an anchor and its overlap in a new probe sample for the palm and wrist BGs. Figure 12.6b, d show an example where the anchor overlap is less than 50%. These are both situations when the anchor has not been found as the anchor just did not exist in the BG. The mean and standard deviation anchor overlap for the palm and wrist databases is shown in column 2 of Table 12.9.

Based on the distribution of anchor overlap in a database, it is possible to choose a minimum value  $O_t$  for the anchor overlap to consider an anchor to be reliable. Choosing a specific  $O_t$  for each database, we measure for each individual, the number of times in the 6 BGs where the anchor is reliably found. This result is shown in column 3 of Table 12.9.

The distributions of anchor overlap and success rates of finding an anchor reliably for both databases is shown in Fig. 12.8. The source code for the anchor-finding algorithms are available at [3].

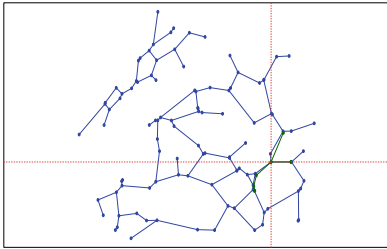
### 12.5.5 Results

Column 1 of Table 12.9 shows that BGs of an individual in the palm database had a greater chance of generating an anchor than BGs of an individual in the wrist database. Anchors are not generated when the BGs from the samples of the individual fail to find a common subgraph among all of them. This happens if even one BG does not have enough common area of capture amongst the six. Figure 12.7a shows an example where 6 BGs from the wrist vein graph could not generate an anchor. Figure 12.7b shows the BGC applied recursively to get a common graph that did not satisfy the two conditions for an anchor, i.e. there was no component of size between 5 and 10 that had at least one claw.

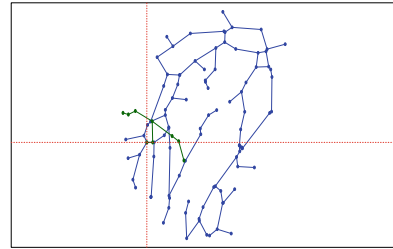
We next wanted to test, if for every failure in getting an anchor, when the selection of BGs changed, would we be able to get an anchor for the individual? We found that out of the 10 individuals whose trials failed to give an anchor in the palm database, only 2 of the individuals failed again when the selection of BGs changed. For the

**Table 12.9** Results from experiments on finding anchors in the PUT palm and wrist databases

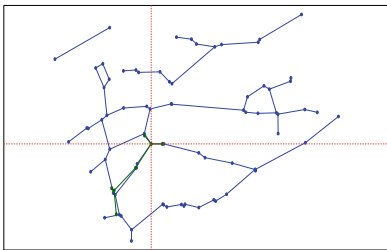
Database PUT	Trials that failed to generate anchors (%)	Anchor overlap across the database (%)	Number of reliable anchor registrations per person
Palm	6	75.17 (16.94)	3.68 (1.96)
Wrist	14	76.6 (14.6)	3.96 (1.64)



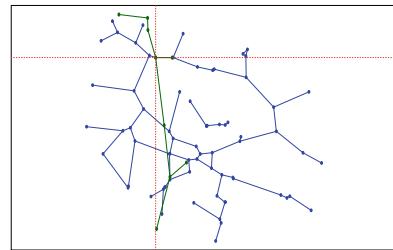
(a) Palm BG with 100% anchor overlap



(b) Palm BG with 44% anchor overlap



(c) Wrist BG with 100% anchor overlap

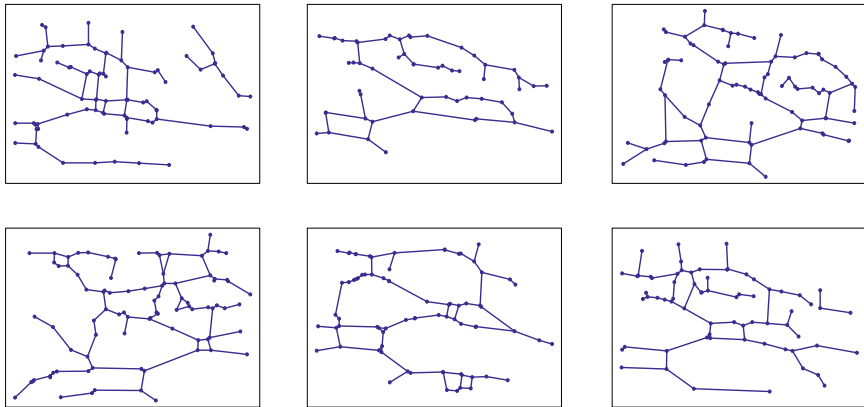


(d) Wrist BG with 44% anchor overlap

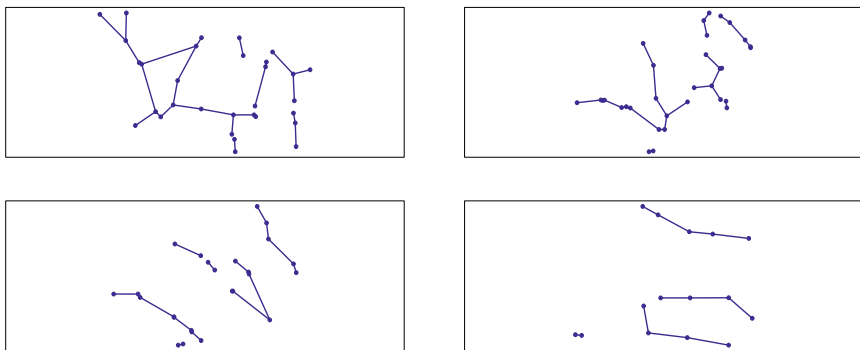
**Fig. 12.6** This figure shows examples of Palm and Wrist BGs where the overlap is 100% (a) and (c), and where the overlap is less than 50% (b) and (d). The anchors are in green and the BGs are in blue

wrist database, 21 individuals failed in a trial to get an anchor, out of them only 3 failed again when the BGs selected were changed. This shows that in practice, if an anchor is not found in a set of samples, it is possible to get an individual to re-enrol until their set of enrolled BGs can give an anchor.

Figure 12.8a, c show the distribution of the anchor overlap measure in the palm and wrist databases. Table 12.9 shows that the mean value of the overlap is over 75% for both. Based on this distribution, we choose  $O_t$  to be 70% and measure the number of times we could reliably find an anchor among the remaining 6 BGs that were not used to get the anchor. Figure 12.8b, d show the distribution of number of times the anchor is found reliably in the remaining samples of an individual in the palm and wrist databases, when  $O_t$  is set to be 70%. Table 12.9 shows that while the palm BGs were more successful overall in finding anchors, once anchors were found, the



(a) Set of 6 wrist BGs that failed to give an anchor



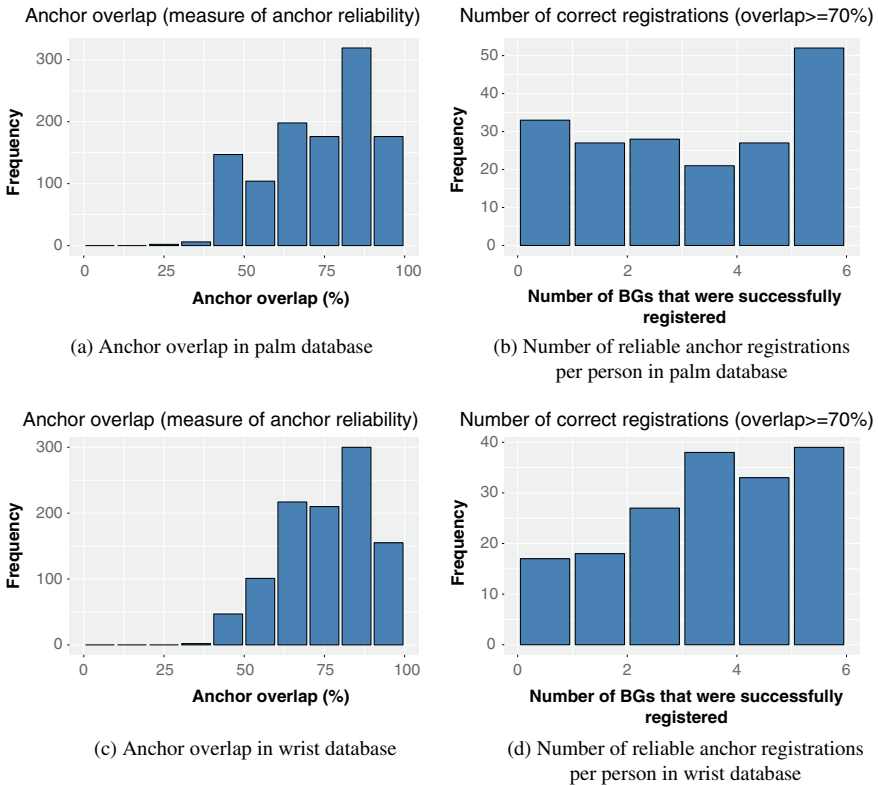
(b) BGC recursively applied to get common graphs

**Fig. 12.7** This figure illustrates how 6 wrist BGs can fail to give an anchor. The final common graph did not have a component of maximum size 10 with at least one claw

wrist BGs had a greater chance of finding the anchor in the remaining BGs from the individual. In practice, it would be possible to request resubmission of the biometric sample if the previously identified anchor wasn't found.

### 12.5.6 Conclusion

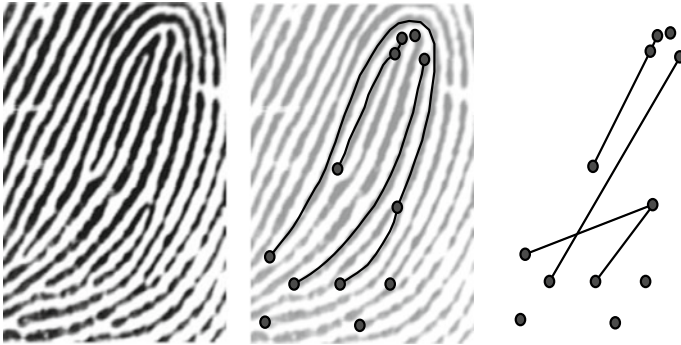
This chapter has explained the basic foundations of representing vascular biometric samples as formal graphs. It has generalised the graph registration and comparison algorithms, BGR and BGC, respectively, and summarised our findings from testing the efficiency and effectiveness of BGR and BGC on 4 different modalities–



**Fig. 12.8** This figure shows the histograms of the anchor overlap in the palm and wrist databases. Once an anchor is found, the number of reliable registrations of the anchor per subject, when  $O_t = 70\%$  is also shown for both databases. Here test set denotes those 6 BGs not used to get the anchor

palm, wrist, hand and retina. The results show that the relational information in BGs provides better recognition accuracy compared to point pattern approaches. We introduced a modification of BGC with the potential to create a template protection scheme using dissimilarity vectors. We also introduced the concept of anchors, a method to register a BG with a consistent reference frame when, unlike retina, there is no intrinsic reference frame. The choice of anchor and structural restrictions are necessary for them to be used to implement biometric template protection using the dissimilarity vector paradigm. We tested the ease of finding anchors and the likelihood for one to be found reliably in BGs that were not used to identify the anchor. The results show us that with proper selection of BGs, we can always find an anchor for an individual.

In the future we want to apply the concept of anchors to test the accuracy of the dissimilarity vector representation for other modalities like palm vein and hand vein. We also plan to conduct a thorough security analysis of the dissimilarity vector rep-



**Fig. 12.9** The extraction of a Biometric Graph from a section of fingerprint image. Note that the BG edges represent the ridgeline connectivity relationships between pairs of minutiae, not the ridgeline itself

resentation as a template protection scheme by establishing empirical and theoretical bounds on the irreversibility and unlinkability of the templates on the lines of work conducted by Gomez et al. [14, 15].

**Acknowledgements** We thank Drs. Hao, Jeffers and Lajervadi for the image processing involved in extracting Biometric Graphs. We thank Dr. Aliahmad and Prof. Kumar for providing the ESRID database. The work has been partly funded by the Australian Research Council grant DP120101188.

## Appendix 1

Here is an example of the original Biometric Graph, as introduced in [17] (Fig. 12.9).

## Appendix 2

This section gives the pseudocode for the BGR algorithm described in Sect. 12.3.1. It is a corrected and updated version of the algorithm in [6]. The source code for the BGR and BGC algorithms is available at [4].

**Require:** Graphs  $g$  and  $g'$  with vertex sets  $\mathbf{V} = \{v_1, v_2, \dots, v_m\}$  and  $\mathbf{V}' = \{v'_1, v'_2, \dots, v'_m\}$  and vertex sets  $\mathbf{E} = \{e_1, e_2, \dots, e_n\}$  and  $\mathbf{E}' = \{e'_1, e'_2, \dots, e'_n\}$ , respectively. Let  $L$  be the number of structure pairs to shortlist and let  $\varepsilon$  be the tolerance to match vertex pairs.

**Ensure:** Aligned graphs  $g_a$  and  $g'_a$  having same edge links as  $g$  and  $g'$  but with new spatial coordinates.

- 1:  $g_a \leftarrow \emptyset$  and  $g'_a \leftarrow \emptyset$ .  $\triangleright$  Initialise the registered graphs that will be returned at the end of the algorithm
- 2:  $\mathbf{S} = \{ s_1, s_2, \dots, s_n \}$  is the list of structures in  $g$ .
- 3:  $\mathbf{S}' = \{ s'_1, s'_2, \dots, s'_{n'} \}$  is the list of structures in  $g'$ .
- 4:  $M_{dist} \leftarrow 0$   $\triangleright$  Initialise a matrix of size  $n \times n'$  with zeros.
- 5: **for**  $a = 1$  to  $n$  **do**
- 6:   **for**  $b = 1$  to  $n'$  **do**
- 7:      $d_{ab} = \text{STRUCTDIST}(s_a, s_b, F)$   $\triangleright$  This function returns the distance between the two structures. The flag  $F$  indicates if the structure is an edge, claw or two-claw.
- 8:      $M_{dist}[a, b] \leftarrow d_{ab}$
- 9:   **end for**
- 10: **end for**
- 11: Sort the contents of  $M_{dist}$  in increasing order.
- 12:  $M_{shortlist}$  is a matrix with 3 columns.  
Every row  $m_i$  stores the 3-tuple  $(d_{abi}, a_i, b_i)$ .  
 $d_{abi}$  is taken from the sorted  $M_{dist}$  with the first row of  $M_{shortlist}$ ,  $m_1$  having  $d_{ab1}$ , the smallest distance.  
 $a_i$  and  $b_i$  indicate the corresponding row and column of  $d_{abi}$  in  $M_{dist}$ .
- 13:  $d_{struct} \leftarrow (0, 0, \dots, 0)_{1 \times L}$   $\triangleright$  A vector to store the distances between graphs when aligned on each of the shortlisted structure pairs
- 14: **for**  $i = 1$  to  $L$  **do**
- 15:    $a = a_i, b = b_i$  where  $m_i \in M_{shortlist}$
- 16:    $g_o = \text{TRANSROT}(g, e_a)$ .  $\triangleright$  Translate and rotate  $g$  with respect to the specific edge in the shortlisted structure
- 17:    $g'_o = \text{TRANSROT}(g', e'_b)$ .
- 18:    $d_{struct}[i] = \text{QUICKSCORE}(g_o, g'_o, \varepsilon)$   $\triangleright$  Compute a distance based on vertex correspondence between the translated and rotated graphs
- 19: **end for**
- 20:  $d_{min} = \text{MIN}(d_{struct})$ .
- 21:  $a_{min}$  and  $b_{min}$  are the row and column in  $M_{shortlist}$  corresponding to  $d_{min}$ .
- 22:  $g_a = \text{TRANSROT}(g, e_{a_{min}})$ .
- 23:  $g'_o = \text{TRANSROT}(g', e'_{b_{min}})$ .  
**return**  $g_a, g'_a$  and  $d_{min}$ .
- 24: **function** EUCDIST( $A = (a_1, a_2, \dots, a_z), B = (b_1, b_2, \dots, b_z)$ )
- 25:    $d = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \dots + (a_z - b_z)^2}$  **return**  $d$
- 26: **end function**
- 27: **function** STRUCTDIST( $s_a, s_b, F$ )  $d_{structPair} \leftarrow \emptyset$
- 28:   **if**  $F ==$  "edge" **then**  $\triangleright$



29:  $E_a \leftarrow (l_a, \theta_a)$   $\triangleright$  The length and slope of the edge

30:  $E_b \leftarrow (l'_b, \theta'_b)$

31:  $d_{structPair} = \frac{1}{0.5(l_a+l'_b)} \text{EUCDIST}(E_a, E_b)$   $\triangleright$  The Euclidean distance between the lengths  $l$  and  $l'$  and slopes  $\theta$  and  $\theta'$  of the vertex pair.

32: **end if**

33: **if**  $F ==$ “claw” **then**

34:  $L_a \leftarrow (l_{1a}, l_{2a}, l_{3a})$   $\triangleright$  The three edges of the claw in decreasing order of edge length

35:  $\Theta_a \leftarrow (\theta_{12a}, \theta_{23a})$   $\triangleright$  The angles between first and second vertex and the second and third vertex.

36:  $L'_b \leftarrow (l'_{1b}, l'_{2b}, l'_{3b})$

37:  $\Theta_b \leftarrow (\theta'_{12b}, \theta'_{23b})$

38:  $l_\delta \leftarrow \text{EUCDIST}(L_a, L'_b)$

39:  $a_\delta \leftarrow \text{EUCDIST}(\Theta_a, \Theta'_b)$

40:  $d = l_\delta + a_\delta$

41:  $d_{structPair} = d$

42: **end if**

43: **if**  $F ==$ “two-claw” **then**  $\triangleright$  A two-claw has two-claw structures connected by a common edge

44:  $L_a \leftarrow (l_{1a}, l_{2a}, l_{3a}, l_{4a}, l_{5a}, l_{6a})$   
 $\triangleright l_1$  and  $l_4$  are the longest edges of the first and second claw structures. The other two edges follow the longest edge in decreasing order of length.

45:  $\Theta_a \leftarrow (\theta_{12a}, \theta_{23a}, \theta_{45a}, \theta_{56a})$   $\triangleright$  The four internal angles, two each from each of the two-claws.

46:  $l_{*a}$  is the length of the connecting edge between the two-claws in structure  $a$  where  $* \in \{1, 2, 3, 4, 5, 6\}$ .

47:  $L'_b \leftarrow (l'_{1b}, l'_{2b}, l'_{3b}, l'_{4b}, l'_{5b}, l'_{6b})$

48:  $\Theta'_b \leftarrow (\theta'_{12b}, \theta'_{23b}, \theta'_{45b}, \theta'_{56b})$

49:  $l'_{*b}$  is the length of the connecting edge between the two-claws in structure  $b$  where  $* \in \{1, 2, 3, 4, 5, 6\}$ .

50:  $d_1 = \text{EUCDIST}(L_a[1 : 3], L'_b[1 : 3]) + \text{EUCDIST}(\Theta_a[1 : 3], \Theta'_b[1 : 3])$

51:  $d_2 = \text{EUCDIST}(L_a[4 : 6], L'_b[4 : 6]) + \text{EUCDIST}(\Theta_a[4 : 6], \Theta'_b[4 : 6])$

52:  $d_3 = \text{EUCDIST}(l_{*a}, l'_{*b})$ .

53:  $d_{structPair} = d_1 + d_2 + d_3$

54: **end if**  
**return**  $d_{structPair}$

55: **end function**

56: **function** TRANSROT( $g, e$ )

57:  $g_o \leftarrow g$

58: The vertex of  $e$  with the smaller  $x$  coordinate will be the origin of the coordinate system.

59: The edge  $e$  will be define the positive direction of the  $x$ -axis.

```

60:   Recalculate all the vertex attributes of  $g_o$  in the new coordinate system.
      return  $g_o$ .
61: end function

62: function QUICKSCORE( $g, g', \varepsilon$ )
63:   Label all vertices of  $g$  and  $g'$  as unmatched.
64:    $C = 0$       ▷ Counter for number of vertex pair matches between  $g$  and  $g'$ 
65:   for  $i = 1$  to  $m$  do
66:     for  $j = 1$  to  $m'$  do
67:       if  $v_i$  is labelled unmatched and  $v'_j$  is labelled unmatched and EU-
          CDIST( $q_i, q'_j$ )  $\leq \varepsilon$  then
68:          $C = C + 1$ .      ▷  $v_i$  matches with  $v'_j$ .
69:         Label  $v_i$  and  $v'_j$  as matched.      ▷  $q_i = (q_{1i}, q_{2i})$  is the vertex
          attribute of  $v_i$  and  $q'_j$  is the vertex attribute of  $v'_j$ .
70:       end if
71:     end for
72:   end for
73:    $d = 1 - \frac{C}{\sqrt{m \times m'}}$ . return  $d$ .
74: end function

```

### Appendix 3

This section presents details of the cost matrices that use complex structures like edges ( $E$ ), claws ( $C$ ) and two-claws ( $T$ ) as structures, as described in Sect. 12.3.2.

#### Edge-based cost matrix:

Let  $u_i, v_i$  be the start and end vertices of  $e_i$  in  $g$  and  $u'_j, v'_j$  be the start and end vertices of  $e'_j$  in  $g'$ . The cost of substituting  $e_i$  with  $e'_j$  given by

$$c_{ij} = ||u_i, u'_j|| + ||v_i, v'_j|| \quad (12.8)$$

where  $||\cdot||$  denotes Euclidean distance between the spatial coordinates of the vertices. The cost of deleting  $e_i$  is

$$c_{i\delta} = \alpha_1 + (\alpha_2 \times (D(u_i) + D(v_i))) \quad (12.9)$$

The cost of inserting  $e'_j$  is

$$c_{\delta j} = \alpha_1 + (\alpha_2 \times (D(u'_j) + D(v'_j))) \quad (12.10)$$

where  $D()$  denotes vertex degree.  $\alpha_1$  denotes the cost for deleting or inserting a vertex.  $\alpha_2$  denotes the cost for deleting or inserting the vertices neighbouring the start and end vertices of the vertex. The cost matrix will have size  $|E| \times |E'|$ , where  $|\cdot|$  denotes cardinality of the set.

**Claw-based cost matrix:**

Let  $c_i$  and  $c'_j$  be the centres of the claws  $s_i$  and  $s'_j$  in  $g$  and  $g'$ . Let  $u_i, v_i, w_i$  and  $u'_j, v'_j, w'_j$  be the end vertices of the three vertices ordered in decreasing order of length for each of the claw structures.

The cost of substituting  $s_i$  with  $s'_j$  given by

$$c_{ij} = ||c_i, c'_j|| + ||u_i, u'_j|| + ||v_i, v'_j|| + ||w_i, w'_j|| \tag{12.11}$$

where  $||\cdot||$  denotes Euclidean distance between the spatial coordinates of the vertices. The cost of deleting  $s_i$  is

$$c_{i\delta} = \alpha_1 + (\alpha_2 \times (D(u_i) + D(v_i) + D(w_i))) \tag{12.12}$$

The cost of inserting  $s'_j$  is

$$c_{\delta j} = \alpha_1 + (\alpha_2 \times (D(u'_j) + D(v'_j) + D(w'_j))) \tag{12.13}$$

where  $D()$  denotes vertex degree.  $\alpha_1$  denotes the cost for deleting or inserting a claw.  $\alpha_2$  denotes the cost for deleting or inserting the vertices neighbouring the end vertices of the claw. The cost matrix will have size  $|S| \times |S'|$ , where  $|\cdot|$  denotes cardinality of the set.

**Two-claw-based cost matrix:**

Let  $t_i$  and  $t'_j$  be two-claw structures in  $g$  and  $g'$ . Each two-claw structures has two-claws connected by a common vertex. Let  $b_i$  and  $c_i$  be the centre vertices of  $t_i$  and  $u_i, v_i, w_i, x_i, y_i, z_i$  be the 6 end vertices of two-claw structures ordered on vertex length.  $u_i$  and  $x_i$  will represent the longest vertices of the claw structures centred on  $b_i$  and  $c_i$ . Similarly let  $b'_j$  and  $c'_j$  represent the centres of the claws and  $u'_j, v'_j, w'_j, x'_j, y'_j, z'_j$  represent the end vertices of the vertices belonging  $t'_j$ . The cost of substituting  $t_i$  with  $t'_j$  given by

$$c_{ij} = ||b_i, b'_j|| + ||c_i, c'_j|| + ||u_i, u'_j|| + ||v_i, v'_j|| + ||w_i, w'_j|| + ||x_i, x'_j|| + ||y_i, y'_j|| + ||z_i, z'_j|| \tag{12.14}$$

where  $||\cdot||$  denotes Euclidean distance between the spatial coordinates of the vertices. The cost of deleting  $t_i$  is

$$c_{i\delta} = \alpha_1 + (\alpha_2 \times (D(u_i) + D(v_i) + D(y_i) + D(z_i))) \tag{12.15}$$

where the  $u_i, v_i, y_i, z_i$  represent the vertices that do not connect the two-claw centres.

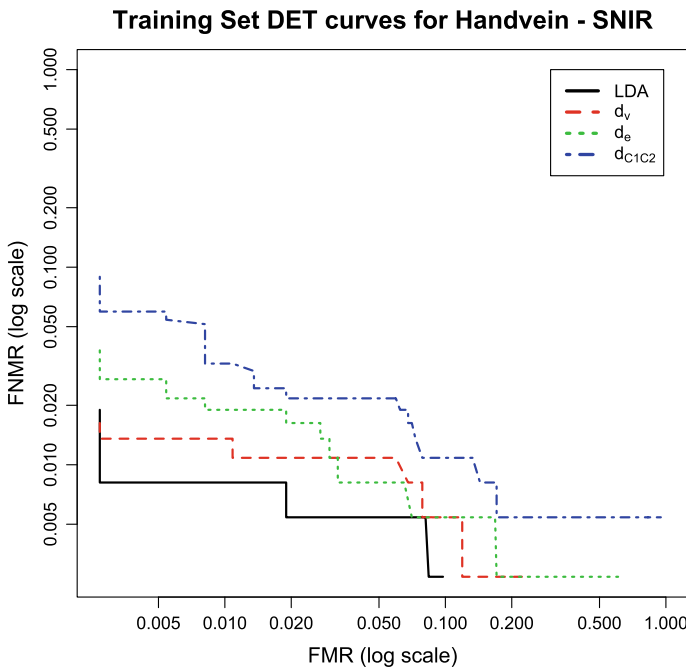
The cost of inserting  $t'_j$  is

$$c_{\delta j} = \alpha_1 + (\alpha_2 \times (D(u'_i) + D(v'_i) + D(y'_i) + D(z'_i))) \tag{12.16}$$

where  $D()$  denotes vertex degree.  $u'_j, v'_j, y'_j, z'_j$  represent the vertices that do not connect  $b'_j$  and  $c'_j$ .  $\alpha_1$  denotes the cost for deleting or inserting a two-claw.  $\alpha_2$  denotes the cost for deleting or inserting the vertices neighbouring the end vertices of the two-claw vertices. The cost matrix will have size  $|T| \times |T'|$ , where  $|\cdot|$  denotes cardinality of the set.

### Appendix 4

In [6] we compared the performance of BGC with standard point pattern based comparison algorithms. Each vascular database was divided into a training and



**Fig. 12.10** DET curve for the top 3 best performing distance measures in the SNIR handvein training dataset. The performance of each distance measure is compared to that obtained when combining the 3 features using an LDA classifier. Results showed that combining the features did not cause a significant improvement in performance over the best performing measure  $d_v$

testing set. The training set was used to determine the best structure for registration, parameters for the graph comparison algorithm and the best distance measure. Once these parameters were picked they were used to test the performance on the testing database at three thresholds corresponding to three specific points from the training database Detection Error Tradeoff (DET) curves—EER, FMR100 and FMR1000. Figure 12.10 shows the DET curves from the SNIR Handvein training dataset. This was not published in [6]. The DETs for all other modalities are available in Fig. 7 in [6].

## References

1. Aastrup Olsen M, Hartung D, Busch C, Larsen R (2011) Convolution approach for feature detection in topological skeletons obtained from vascular patterns. In: 2011 IEEE workshop on computational intelligence in biometrics and identity management (CIBIM), pp 163–167 (2011)
2. Aliahmad B, Kumar DK, Hao H, Kawasaki R (2013) Does fractal properties of retinal vasculature vary with cardiac cycle? In: 2013 ISSNIP biosignals and biorobotics conference: biosignals and robotics for better and safer living (BRC), pp 1–4 (2013). <https://doi.org/10.1109/BRC.2013.6487480>
3. Arakala A. Source code for anchors for graph registration. [https://github.com/ArathiArakalaRMIT/anchor\\_dissimilarityVector](https://github.com/ArathiArakalaRMIT/anchor_dissimilarityVector)
4. Arakala A. Source code for biometric graph matching for vascular biometrics. [https://github.com/ArathiArakalaRMIT/BGM\\_for\\_vascular\\_graphs](https://github.com/ArathiArakalaRMIT/BGM_for_vascular_graphs)
5. Arakala A, Davis SA, Hao H, Aliahmad B, Kumar DK, Horadam KJ (2019) Template protected biometrics using dissimilarity vectors (preprint 2019)
6. Arakala A, Davis SA, Hao H, Horadam KJ (2017) Value of graph topology in vascular biometrics. *IET Biom* 6(2):117–125. <https://doi.org/10.1049/iet-bmt.2016.0073>
7. Arakala A, Davis SA, Horadam KJ (2011) Retina features based on vessel graph substructures. In: 2011 international joint conference on biometrics (IJCB), pp 1–6 (2011). <https://doi.org/10.1109/IJCB.2011.6117506>
8. Arakala A, Hao H, Davis S, Horadam K (2015) The palm vein graph-feature extraction and matching. In: Proceedings of the first international conference on information systems security and privacy (ICISSP), pp 56–64. Loire Valley, France. <https://doi.org/10.5220/0005239102950303>
9. Bunke H, Shearer K (1998) A graph distance metric based on the maximal common subgraph. *Pattern Recogn Lett* 19(3–4):255–259. [https://doi.org/10.1016/S0167-8655\(97\)00179-7](https://doi.org/10.1016/S0167-8655(97)00179-7)
10. Chen H, Lu G, Wang R (2009) A new palm vein matching method based on ICP algorithm. In: Proceedings of the 2nd international conference on interaction sciences: information technology, culture and human, ICIS '09, pp 1207–1211. ACM, New York, NY, USA. <https://doi.org/10.1145/1655925.1656145>. URL <http://doi.acm.org.ezproxy.lib.rmit.edu.au/10.1145/1655925.1656145>
11. Davis S, Abbasi B, Shah S, Telfer S, Begon M (2014) Spatial analyses of wildlife contact networks. *J R Soc Interface* 12(102). <https://doi.org/10.1098/rsif.2014.1004>. <http://rsif.royalsocietypublishing.org/content/12/102/20141004>

12. Deng K, Tian J, Zheng J, Zhang X, Dai X, Xu M (2010) Retinal fundus image registration via vascular structure graph matching. *J Biomed Imaging* 2010(14):1–13. <https://doi.org/10.1155/2010/906067>
13. Drechsler K, Laura CO (2010) Hierarchical decomposition of vessel skeletons for graph creation and feature extraction. In: 2010 IEEE international conference on bioinformatics and biomedicine (BIBM), pp 456–461. <https://doi.org/10.1109/BIBM.2010.5706609>
14. Gomez-Barrero M, Galbally J, Rathgeb C, Busch C (2018) General framework to evaluate unlinkability in biometric template protection systems. *IEEE Trans Inf Forensics Secur* 13(6):1406–1420. <https://doi.org/10.1109/tifs.2017.2788000>. <https://app.dimensions.ai/details/publication/pub.1100133741>. <https://app.dimensions.ai>. Accessed 28 Feb 2019
15. Gomez-Barrero M, Rathgeb C, Galbally J, Busch C, Fierrez J (2016) Unlinkable and irreversible biometric template protection based on bloom filters. *Inf Sci* 370:18–32. <https://doi.org/10.1016/j.ins.2016.06.046>. <https://app.dimensions.ai/details/publication/pub.1052550524>. <https://app.dimensions.ai>. Accessed 28 Feb 2019
16. Gouri R (2013) A vascular network matching algorithm for physiological face recognition. Master's thesis, Houston, TX, USA
17. Horadam KJ, Davis SA, Arakala A, Jeffers J (2011) Fingerprints as spatial graphs-nodes and edges. In: Proceedings of international conference on digital image computing techniques and applications, DICTA, pp 400–405. Noosa, Australia
18. Kabacinski R, Kowalski M (2011) Vein pattern database and benchmark results. *Electron Lett* 47(20):1127–1128
19. Khakzar M, Pourghassem H (2017) A retinal image authentication framework based on a graph-based representation algorithm in a two-stage matching structure. *Biocybern Biomed Eng* 37(4):742–759. <https://doi.org/10.1016/j.bbe.2017.09.001>
20. Lajevardi S, Arakala A, Davis S, Horadam K (2013) Retina verification system based on biometric graph matching. *IEEE Trans Image Process* 22(9):3625–3635
21. Lajevardi S, Arakala A, Davis S, Horadam K (2014) Hand vein authentication using biometric graph matching. *IET Biom* 3(4):302–313. <https://doi.org/10.1049/iet-bmt.2013.0086>
22. Lupacu CA, Tegolo D, Bellavia F, Valenti C (2013) Semi-automatic registration of retinal images based on line matching approach. In: Proceedings of the 26th IEEE international symposium on computer-based medical systems, pp 453–456. <https://doi.org/10.1109/CBMS.2013.6627839>
23. Nibbelke V (2017) Vascular pattern recognition for finger veins using biometric graph matching. Master's thesis, Twente, Netherlands
24. Ortega M, Penedo MG, Rouco J, Barreira N, Carreira MJ (2009) Retinal verification using a feature points-based biometric pattern. *EURASIP J Adv Signal Process* 2009:1–13
25. Rathgeb C, Busch C (2012) Multi-biometric template protection: issues and challenges. In: Yang J, Xie SJ (eds) *New trends and developments in biometrics* (chap. 8). IntechOpen, Rijeka. <https://doi.org/10.5772/52152>
26. Riesen K, Bunke H (2010) *Graph classification and clustering based on vector space embedding*, 1st edn. World Scientific
27. Wang L, Leedham G, Cho SY (2007) Infrared imaging of hand vein patterns for biometric purposes. *IET Comput Vis* 1(3–4):113–122

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

