

CONCURSO
DE TESIS
DE GRADO

DC·FCEN·UBA

Edición 2016

Prefacio

Este volumen contiene los trabajos presentados al 4to Concurso de Tesis de Licenciatura del Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires. Todos los trabajos presentados fueron evaluados por profesionales destacados de una gran lista de universidades nacionales y han mostrado una calidad excepcional que se vio en las evaluaciones. De los trabajos presentados fueron elegidos tres como ganadores y otros tres a los que se les otorgó una mención especial. Al mismo tiempo, los trabajos ganadores fueron presentados en una sesión de charlas el día 21 de julio de 2016 mientras que los trabajos con mención especial fueron presentados en formato de poster el día 20 de julio de 2016, ambos eventos en el marco de la Escuela de Ciencias Informáticas 2016.

Desde la Secretaría de Investigación del Departamento de Computación de la Facultad de Ciencias Exactas y Naturales de la Universidad de Buenos Aires queremos agradecer enormemente la colaboración de autores, directores, autoridades de la institución, miembros del comité evaluador, revisores externos y especialmente a la organización de la escuela por haber hecho este concurso posible. Para terminar, agradecemos EasyChair por haber proveído el soporte técnico necesario para realizar el evento.

Felicitaciones a los participantes del concurso y sus directores.

23 de junio de 2016
Ciudad Autónoma de Buenos Aires, Argentina

Carlos Gustavo Lopez Pombo & Flavia
Bonomo

Comité de Programa

Andrés Diaz-Pace

Raul Fervari

Pablo Fillottrani

Alejandra Garrido

Carlos Gustavo Lopez Pombo

Pablo E. Martinez-Lopez

Javier Martinez-Viademonte

Germán Regis

Dante Zanarini

ISISTAN Research Institute, UNICEN University

FaMAF - Universidad Nacional de Córdoba

Universidad Nacional del Sur

LIFIA-Facultad de Informatica-UNLP

Department of Computing, Facultad de Ciencias Exactas y Naturales,

Universidad de Buenos Aires and CONICET

UNQ

Universidad Nacional de General Sarmiento

Universidad Nacional de Río Cuarto

Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas

Revisores Adicionales

Luque, Franco M.
Moscato, Mariano
Rosner, Nicolás
Sánchez, Jorge
Vidal, Santiago

Tabla de Contenidos

Predicción de los Rendimientos de Acciones en Argentina Sobre la Base de Indicadores Técnicos y el Modelado de Tópicos de un Foro Online.....	1
<i>Ramiro H. Gálvez</i>	
Reconocimiento de Objetos en Imágenes RGBD	3
<i>Nadia Mariel Heredia Favieri y Francisco Gomez Fernandez</i>	
Implementación de un runner de Gobstones para la plataforma Mumuki	5
<i>Federico Aloí</i>	
Métricas de mimetización acústico-prosódica en hablantes y su relación con rasgos sociales de diálogos	7
<i>Juan Manuel Pérez</i>	
Un asistente flexible para la formación de equipos en base a técnicas SAT	19
<i>Ignacio Gatti</i>	
Detección y cierre de ciclos en sistemas SLAM basados en visión estéreo	32
<i>Gastón Ignacio Castro, Pablo De Cristóforis y Taihú Pire</i>	

Predicción de los rendimientos de acciones en Argentina sobre la base de indicadores técnicos y el modelado de tópicos en foros bursátiles

Ramiro H. Gálvez
rgalvez@dc.uba.ar

Resumen de Tesis de Maestría
Maestría en Explotación de Datos y Descubrimiento del Conocimiento
Facultad de Ciencias Exactas y Naturales, Facultad de Ingeniería
Universidad de Buenos Aires, Argentina

1. Resumen de tesis

Sobre posts registrados en el foro online de Rava Bursátil¹ aplicamos técnicas provenientes del procesamiento del lenguaje natural para detectar la presencia de tópicos subyacentes en las conversaciones que se realizan en un día. Una vez hecho esto, intentamos responder dos preguntas. Primero, ¿tienen estos tópicos información predictiva referida al retorno futuro de una acción? (P1). Segundo, de haber información predictiva, ¿es la misma novedosa? (P2).

Utilizamos datos de dos fuentes. Primero, obtenemos la evolución de las cotizaciones diarias de las acciones que cotizan en el Merval. Segundo, recolectamos datos de un foro bursátil en línea.² El foro está compuesto por TEMAS identificados por un título. Cada tema está compuesto a su vez por POSTS, los cuales contienen los comentarios que los usuarios hacen en el foro (los mismos pueden incluir EMOJIS, citas a otros comentarios anteriores e imágenes). Cada post contiene metadata referida a cuándo fue escrito (fecha y hora) y a quién lo escribió.

Se procesan y normalizan los posts diarios que pertenecen a los distintos temas, utilizando una variante del modelo de bolsa de palabras. Luego se lleva adelante un proceso de reducción de dimensionalidad (descomposición SVD truncada). Como resultado de este proceso se obtiene una serie de atributos que, presumimos, tienen contenido semántico y pueden ser asociados a tópicos que se debaten.

Como en este trabajo realizamos predicciones sobre datos con estructura de series de tiempo, optamos por utilizar el esquema de VENTANAS CRECIENTES [3] para calibrar, validar y evaluar los experimentos. Las técnicas que empleamos deben calibrarse para que obtengan buenos resultados en el conjunto de testeo; para lograr esto, nuestros experimentos tienen la siguiente estructura. Para cada acción se dividen los datos en tres conjuntos: CONJUNTO DE ENTRENAMIENTO, CONJUNTO DE VALIDACIÓN y CONJUNTO DE TESTEO. Luego, se prueban distintas variantes de los parámetros sobre los datos de entrenamiento, midiendo la

¹ Véase <http://foro.ravaonline.com>.

² Véase <http://foro.ravaonline.com/foro3/index.php>.

performance sobre el conjunto de validación y utilizando el esquema de ventanas crecientes. Se elige como “óptimos” a los parámetros que dieron lugar al máximo desempeño en función de una métrica de performance elegida.

Para responder a P1 entrenamos modelos que predicen el rendimiento que tendrá una acción en un día t sobre la base de información contenida en los primeros h tópicos detectados del foro. Elegimos RIDGE REGRESSION [1] como modelo de aprendizaje a usar en este experimento. Comparamos los resultados contra los que obtiene un modelo que predice, para cada día, un valor igual al promedio de los rendimientos para el conjunto de entrenamiento. Adicionalmente, comparamos los resultados con los por un modelo que predice como rendimiento de una acción para un día determinado el que tuvo la acción en el día previo ($\hat{r}_t = r_{t-1}$).

Para responder P2 replicamos modelos desarrollados en la literatura de aprendizaje automático y vemos cómo se modifica su performance cuando se les agrega información proveniente de la interacción de los participantes del foro en línea. Estos modelos se caracterizan por utilizar indicadores técnicos como variables predictoras. Los indicadores técnicos son medidas que se arman sobre la base de datos pasados de cotización de las acciones y que se presume dan indicios sobre el comportamiento futuro que tendrán los retornos. La elección de qué indicadores técnicos utilizar en nuestros modelos surge de replicar los modelos presentados en [2]. En este experimento en vez de predecir el valor de r_t nos enfocamos en predecir su dirección ($rc_t = 1$ si $r_t \geq 0$ y $rc_t = 0$ en caso contrario). Elegimos RANDOM FOREST [1] como la técnica utilizada para predecir en este experimento. Para evaluar la performance de los sistemas entrenados utilizamos dos métricas: Accuracy y AUC.

Obtenemos respuestas afirmativas a las dos preguntas principales de este trabajo. En lo referido a P1, nuestros resultados que indican que sí es posible extraer dicha información, pues modelos que predicen retornos diarios de acciones únicamente sobre la base de tópicos obtienen resultados al nivel o incluso mejores a los de dos modelos rivales. En lo referido a P2, al estudiar si modelos de aprendizaje automático basados en el uso de indicadores técnicos mejoran su performance al incorporar los tópicos detectados del foro, obtenemos resultados que indican que esto efectivamente ocurre. A su vez, la metodología propuesta efectivamente detecta tópicos que poseen contenido semántico.

Referencias

1. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.
2. Yakup Kara, Melek Acar Boyacioglu, and Ömer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the istanbul stock exchange. *Expert systems with Applications*, 38(5):5311–5319, 2011.
3. Luís Torgo. *Data Mining with R, learning with case studies*. Chapman and Hall/CRC, 2010.

Reconocimiento de Objetos en Imágenes RGBD

Nadia Mariel Heredia Favieri¹ y Francisco Gómez-Fernández¹

Universidad de Buenos Aires {nheredia, fgomez}@dc.uba.ar

Resumen Implementamos un sistema de reconocimiento de objetos, instanciable con distintos métodos por etapa, con datos de sensores RGB-D estilo Kinect. Experimentamos variando cada etapa del sistema, los parámetros y métodos, observando la influencia en el desempeño y los tiempos. Usar más puntos de muestreo otorga mayores precisiones pero peores tiempos, y disminuir los radios de los descriptores y las normales acelera la generación de descriptores pero ralentiza la búsqueda de correspondencias.

Analizamos los descriptores FPFH, SHOT y Spin-Images, y estudiamos diferentes distancias para comparar descriptores, y métodos para establecer correspondencias. El rendimiento del sistema se considera en dos bases de datos con ground truth, la primera un caso ideal y la segunda un caso real. En el caso ideal, todos los descriptores otorgan buen desempeño, siendo Spin-Images la mejor opción. Para el caso real, FPFH es muy lento, y SHOT y Spin-Images ofrecen un buen compromiso entre exactitud y tiempo.

Keywords: RGB-D · object recognition · point cloud · descriptor

1. Sistema Implementado

En este trabajo analizamos el comportamiento de un sistema de reconocimiento en una aplicación real, usando imágenes RGB-D, con información de color y de profundidad de una escena provistas por sensores como Kinect.

Representamos los datos a lo largo de este trabajo con nubes de puntos, colecciones de puntos tridimensionales. El sistema de reconocimiento implementado está dividido en tres etapas: 1. extracción de keypoints 2. elaboración de descriptores 3. búsqueda de correspondencias (matching). Usamos tres descriptores basados en histogramas: *SHOT* [5], *FPFH* [4], y *Spin-Images* [1], que capturan información de la superficie subyacente en formas diferentes. Todos estos descriptores en su elaboración utilizan la totalidad de la nube de puntos, pero sólo se computan para los puntos seleccionados.

Usamos dos bases de datos para evaluar el sistema construido, la base de objetos RGB-D, tomada de Kevin Lai et al. [2] presentando un caso real, y la base de CV-Lab, tomada del trabajo de Tombari et al. [5]. presentando un caso ideal, ambas con información de *ground-truth*. Para obtener los keypoints, submuestreamos uniformemente cada 1cm en el modelo, cada 2cm en las escenas de la base CVLab, y 1cm en las de RGB-D. Para la base CVLab usamos un radio

de 1cm para las normales y de 2cm para los descriptores, y para la de RGB-D, un radio de 2cm para las normales y de 5cm para los descriptores. Para el matching, usamos el método del cociente [3] con un umbral de 0.90 para todos los descriptores.

2. Resultados Principales

Vimos que el uso de un submuestreo uniforme es más rápido que usar el detector de keypoints de Harris, y resulta en más puntos, además un tamaño más chico de submuestreo otorga mejores precisiones a costa de mayores tiempos. Reducir el radio de los descriptores y normales disminuye el tiempo de generación de descriptores pero incrementa el tiempo de búsqueda de correspondencias. La densidad de las nubes y la cantidad de puntos elegidos afectan fuertemente el rendimiento, modificando la distribución de tiempos de las etapas y la precisión final. Cuanto mayor sea la densidad, más información se incorpora en los descriptores, haciendo que el reconocimiento sea mucho más preciso pero que demore más por la cantidad extra de información que se utiliza para estimar las normales y elaborar descriptores.

En la base de CV-Lab [5] el sistema otorga buenos valores de precisión y tiempos bajos con todos los descriptores. Sin embargo en la base RGB-D [2], solo se logran buenos resultados usando los descriptores CSHOT y SI. En particular la mayoría de los objetos de esta base no tienen mucha textura, haciendo que el uso de la información de color no otorgara mejoras, y que la información de forma sea muy importante. SI que usa solo forma resulta por lo tanto ser el mejor candidato. Por otro lado, en los casos donde sí existe textura, el rendimiento de CSHOT resulta mejor que el de SI.

Referencias

1. Johnson, A.E.: Spin-images: A Representation For 3-D Surface Matching. Ph.D. thesis, The Robotics Institute, Carnegie Mellon University (1997)
2. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: Robotics and Automation (ICRA), 2011. IEEE International Conference on. pp. 1817–1824. IEEE (2011)
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)* 60(2), 91–110 (2004)
4. Rusu, R.B., Blodow, N., Beetz, M.: Fast point feature histograms (FPFH) for 3D registration. In: Robotics and Automation (ICRA), 2009. IEEE International Conference on. pp. 3212–3217. IEEE (2009)
5. Tombari, F., Salti, S., Di Stefano, L.: Unique signatures of histograms for local surface description. In: European Conference on Computer Vision (ECCV). pp. 356–369. Springer (2010)

Implementación de un runner de Gobstones para la plataforma Mumuki

Federico Aloí¹

Universidad Nacional de Quilmes,
federico.aloi@gmail.com

Keywords: educación, enseñanza de la programación, análisis estático de código, corrección automática de programas

1. Resumen

Muchos cursos iniciales se siguen dictando usando lenguajes de bajo nivel como soporte, haciendo énfasis en cuestiones que tienen más que ver con el *hardware* subyacente que con el *software* que se pretende construir: se favorece a la performance del programa por sobre otros atributos como expresividad, abstracción, declaratividad. Este enfoque no resulta adecuado para un primer acercamiento a la programación y en muchos casos lo único que se logra es ahuyentar a los estudiantes. Analizar las causas de este fenómeno escapa al objetivo de este trabajo.¹

En contraposición, en los últimos tiempos han cobrado relevancia otras formas de enseñar a programar que pretenden ser más inclusivas y didácticas, haciendo hincapié en el *pensamiento computacional*[1]. Para lograrlo, hacen uso intensivo de herramientas tecnológicas; el presente trabajo se concentró en dos de ellas: Gobstones y Mumuki, considerando que pueden complementarse muy bien, logrando enriquecer aún más el método de enseñanza de un curso introductorio en programación.

1.1. Gobstones, una Nueva Forma de Aprender a Programar

El proyecto Gobstones consta de dos componentes fundamentales: una secuencia didáctica y un lenguaje de programación que permite ponerla en práctica.

La secuencia se diseñó teniendo en cuenta el trabajo con los estudiantes de primer año de la Tecnicatura en Programación Informática de la UNQ y favorece la abstracción desde el primer momento, prestando especial atención al uso de procedimientos y funciones y postergando otros conceptos, como el uso de variables, hasta que realmente surge la necesidad de utilizarlos.

¹ Para conocer más al respecto, se recomienda la lectura del documento *Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas*, publicado por la Fundación Sadosky.

II REFERENCIAS

En cuanto al lenguaje, no es posible realizar operaciones de entrada/salida tradicionales; el contacto con el “mundo exterior” se da a través de un tablero de dimensiones configurables, que puede contener bolitas de colores en cada una de sus celdas. El autómata que ejecuta sobre el tablero es denominado *cabezal*; sus operaciones básicas son: desplazarse por el tablero, poner, sacar y censar bolitas.

1.2. Mumuki, Educación Libre de la Programación

Mumuki es un proyecto que apunta a universalizar el acceso a la educación libre, gratuita y de calidad, enfocándose principal pero no excluyentemente en la enseñanza de la programación. Según sus fundadores es, además,

“un software educativo para aprender a programar a partir de la resolución de problemas; plantea enseñar conceptos de programación, en un proceso conducido por guías prácticas en las que la teoría surge a medida que se avanza. Esta herramienta se presenta al estudiante como una aplicación Web interactiva, en la que se articulan explicaciones y ejemplos con la opción de que cada uno realice su propia solución y la plataforma la pruebe y corrija instantáneamente, orientando acerca de los aciertos y errores.”[3]

La plataforma es extensible, en tanto permite incorporar nuevos lenguajes que pueden ser usados como medio para escribir ejercicios. Esto permitió la construcción de una integración que permite ejecutar y validar programas Gobstones dentro de la plataforma.

1.3. Conclusiones

Si bien todavía faltan funcionalidades para considerar que la integración está completa, el producto construido resulta funcional y deja planteadas varias líneas de trabajo para continuar su desarrollo. La versión actual se encuentra productiva y está siendo utilizada por estudiantes de nivel medio y superior, con una base de más de cien ejercicios.

Pueden fácilmente escribirse nuevos ejercicios y extender el uso de la plataforma a cualquier asignatura que utilice Gobstones como soporte para la enseñanza de la programación.

Referencias

- [1] Jeannette M. Wing. «Computational thinking». En: *Communications of the ACM* 49, No. 3 (2006), págs. 33-35.
- [2] Fundación Sadosky. *Una propuesta para refundar la enseñanza de la computación en las escuelas Argentinas*. 2013.
- [3] Federico Aloí, Franco Bulgarelli y Lucas Spigariol. «Mumuki, una plataforma libre para aprender a programar». En: *Memorias CONAIISI*. 2015.

Métricas de mimetización acústico-prosódica en hablantes y su relación con rasgos sociales de diálogos

Juan Manuel Pérez
jmperez@dc.uba.ar

Tesis de Licenciatura en Ciencias de la Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Resumen El *entrainment* (mimetización) es un fenómeno inconsciente que se manifiesta a través de la adaptación de posturas, forma de hablar, gestos faciales y otros comportamientos entre dos o más interactores, fuertemente emparentado con el sentimiento de empatía entre los participantes. En el presente trabajo, nos proponemos explorar y refinar una métrica del entrainment acústico-prosódico definida en trabajos previos. Analizamos la relación entre los valores obtenidos y las percepciones sociales que terceros tienen sobre las conversaciones, en un corpus de diálogos orientados a tareas en inglés.

Keywords: Procesamiento del Habla, Series de Tiempo, Entrainment

1. Introducción

Los sistemas de diálogo humano-computadora son cada vez más frecuentes, y sus aplicaciones comprenden una amplia gama de rubros: desde aplicaciones móviles, motores de búsqueda, juegos o tecnologías de asistencia para ancianos y discapacitados. Si bien es cierto que estos sistemas logran captar la dimensión lingüística de la comunicación humana, tienen un déficit importante a la hora de procesar y transmitir el aspecto superestructural de la comunicación oral, que radica en el intercambio de afecto, emociones, actitudes y otras intenciones de los participantes. Este problema puede verse en cualquier sistema que interactúe sintetizando lenguaje humano, como por ejemplo, las aplicaciones telefónicas que atienden automáticamente a sus clientes [16]. Stanley Kubrick y Arthur C. Clarke predijeron esto a la perfección, cuando en “2001: Una Odissea en el Espacio” (1968) dotaron a *HAL* de una voz monótona y robótica, casi lobotomizada.

Dentro de las cualidades del lenguaje oral, una de las más distintivas es la *prosodia*, que es la dimensión que capta *cómo* se dicen las cosas, en contraposición a *qué* se está manifestando. Posee varias componentes acústico-prosódicas: por ejemplo, el tono o pitch, la intensidad o volumen, la calidad de la voz, la velocidad del habla. Un manejo adecuado de estas componentes es lo que, hoy

día, distingue una voz humana de una artificial. Esta carencia de habilidad sobre la prosodia conlleva cierta dificultad en la interacción con agentes conversacionales, que suelen ser calificados como “mecánicos” o “extraños” en su forma de comunicarse [19].

En pos de mejorar el entendimiento entre agentes conversacionales y sus usuarios, resulta de vital importancia poder entender y modelar las variaciones prosódicas de la comunicación oral. Esto se traduciría tanto en una mejor apreciación de lo que quiere comunicar el usuario, como en una mayor naturalidad de la voz sintetizada por el agente.

1.1. Mimetización

Se ha observado que, bajo ciertas condiciones, cuando una persona mantiene una conversación modifica su manera de actuar aproximándola a la de su interlocutor. En una reseña de este tema se describe a este fenómeno como una “imitación no consciente de posturas, maneras, expresiones faciales y otros comportamientos del compañero interaccional” [4, p. 893] y conjeturan que es más fuerte en individuos con empatía disposicional. En otras palabras, personas con predisposición a buscar la aceptación social modifican su comportamiento en forma más marcada para aproximarlos a sus interlocutores.

Esta modificación del comportamiento ha sido observada también en la manera de hablar. Por ejemplo, los interlocutores adoptan las mismas formas léxicas para referirse a las cosas, negociando tácitamente descripciones compartidas, en especial para cosas que resulten poco familiares [2]. Estudios más recientes sugieren que esto también es cierto para el uso de estructuras sintácticas [18]. Este fenómeno subconsciente es conocido como mimetización, alineamiento, adaptación o convergencia y también con el término inglés *entrainment* y se ha mostrado que juega un rol importante en la coordinación de diálogos, facilitando tanto la producción como la comprensión del habla en los humanos [9]. En nuestro caso, nos interesa principalmente el *entrainment* de la prosodia.

1.2. Midiendo la mimetización

Muchos estudios han examinado la mimetización prosódica, listados en De Looze et al(2013) [7]. Un número importante de ellos se han basado en la premisa de la mimetización como un fenómeno lineal, en el cual la convergencia “va sucediendo” a lo largo de la conversación [3]. Estos estudios dividen las conversaciones en varias partes, y verifican que la diferencia absoluta entre los valores medios (de las variables acústico-prosódicas) y sus desviaciones se aproxime en las últimas partes de la interacción. Sin embargo, este enfoque de la mimetización niega su faceta dinámica: los interlocutores pueden estar inactivos y luego hablar, pueden pasar por varias etapas como escuchar, pensar, discutir un punto, etc. En el trabajo de Levitan et al [15] se reportó que éste es un fenómeno no solamente lineal, sino también dinámico, donde los interlocutores van coincidiendo en el análisis por turnos.

Un problema común que surge a la hora de calcular estas métricas es el hecho de que las conversaciones no están alineadas en el tiempo, ni se dan en turnos de duración constante. Nos preguntamos entonces qué partes del diálogo de un hablante deberían compararse con qué otras partes de su par. Un enfoque de comparar interlocuciones uno a uno es demasiado simple y no captura situaciones de diálogo reales, mucho más dinámicas y con solapamiento casi constante.

Para atacar estos inconvenientes, utilizamos el método *TAMA* (Time Aligned Moving Average) desarrollado por Kousidis et al [13], que consiste en separar en ventanas de tiempo el diálogo, y promediar los valores de las variables prosódicas dentro de cada una. Este método es muy similar a aplicar un filtro de Promedio Móvil (Moving Average), lo que da el nombre a la técnica. Al separar el diálogo en ventanas de tiempo, podemos construir dos series de tiempo en base a cada interlocutor. Estas abstracciones son mucho más tratables que tener una secuencia de elocuciones de parte de cada hablante, y nos permiten efectuar análisis bien conocidos, uno de los cuáles nos permite construir una medida del *entrainment*.

En este trabajo, definimos dos métricas de *entrainment* basadas en *TAMA* y las aplicamos sobre un corpus de diálogo entre dos participantes angloparlantes, quienes interactúan mediante un juego a través de computadoras. El corpus ha sido anotado manualmente con variables que describen la percepción social de la conversación; por ejemplo: ¿el sujeto parece comprometido con el juego? ¿al sujeto no le agrada su compañero?

Luego veremos si existe, para cada una de las variables acústico-prosódicas, alguna relación significativa entre las métricas definidas y las percepciones sociales sobre las conversaciones. Uno esperaría que valores altos de nuestras métricas del *entrainment* se relacionen con valores altos de variables sociales positivas, tales como mostrarse colaborativo o compenetrado en la tarea.

2. Materiales y Método

2.1. Columbia Games Corpus

Empleamos para este estudio el Columbia Games Corpus [8] que consiste de doce conversaciones diádicas entre trece personas angloparlantes distintas. Todos los participantes reportaron hablar inglés americano estándar, y no tener problemas de audición.

Cada sesión duró aproximadamente 45 minutos, totalizando 9 horas de diálogos, 70.259 palabras (2.037 únicas) para todo el cuerpo de datos. Todas las conversaciones cuentan con transcripciones textuales alineadas temporalmente a la señal de audio, realizadas por personal especialmente entrenado.

En cada sesión, se sentó a dos participantes (quienes no se conocían previamente) en una cabina profesional de grabación, cara a cara a ambos lados de una mesa, y con una cortina opaca colgando entre ellos para evitar la comunicación visual. Los participantes contaron con sendas computadoras portátiles conectadas entre sí, en las cuales jugaron una serie de juegos simples que requerían de comunicación verbal. El primero de ellos es un juego de cartas que no

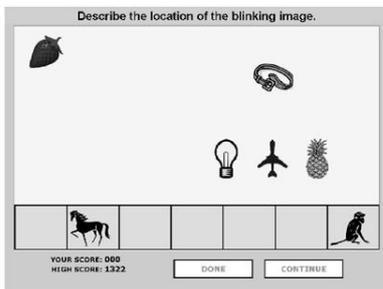


Figura 1. Juego de objetos del Columbia Games

consideramos en el presente estudio por tratarse esencialmente de monólogos o diálogos con poca interacción. Luego de esto, pasaron al juego que analizamos, denominado ‘juego de objetos’.

En el juego de objetos, la pantalla de cada jugador mostró un tablero con varios objetos, entre 5 y 7, como se ve en la Figura 1. Para uno de los jugadores (el Descriptor) el objeto *Objetivo* aparecía en una posición aleatoria entre otros objetos. Para el otro jugador, a quien llamaremos el Seguidor, el objetivo aparecía en la parte baja de la pantalla. Entonces, al Descriptor se le encargaba describir la posición del Objetivo de manera que el Seguidor pudiera mover su representación del objeto a la misma posición en su pantalla. Luego de una negociación entre ambos jugadores para decidir la mejor posición del objeto, se les asignó a los jugadores una puntuación entre 1 y 100 puntos de acuerdo a qué tan acertado fue el posicionamiento del objetivo por parte del Seguidor.

Cada sesión consistió de 14 tareas como ésta, cambiando los objetos y sus ubicaciones. En las primeras cuatro tareas, uno de los sujetos tomó el papel del Descriptor; en los siguientes cuatro invirtieron roles, y en las finales seis fueron alternando los roles de Descriptor y Seguidor.

2.1.1. Anotaciones sociales y acústico/prosódicas Varios aspectos del comportamiento de los jugadores durante los juegos de objetos fueron anotados mediante la herramienta de crowdsourcing *Amazon Mechanical Turk*¹. Cada anotador escuchó el audio correspondiente a una tarea del juego y tuvo que responder a varias preguntas, listadas en la Figura 1

Cada uno de estos audios fue puntuado por cinco anotadores, que respondieron por sí o por no para cada una de las preguntas. El puntaje que recibe cada una de las preguntas (a las cuales llamaremos a partir de ahora *variables sociales*) consiste en la cantidad de respuestas afirmativas que recibió, teniendo un rango de 0 a 5.

A su vez, los valores de las variables acústico-prosódicas listadas en la Figura 2 fueron medidos para cada segmento del habla del corpus. Para entender mejor a qué se refieren estas variables, repasamos a continuación algunos conceptos:

¹ <https://www.mturk.com>

Nombre	Pregunta
<i>contributes-to-completion</i>	¿el sujeto contribuye para el éxito del equipo?
<i>engaged-with-game</i>	¿el sujeto parece comprometido con el juego?
<i>making-self-clear</i>	¿el sujeto se expresa correctamente?
<i>planning-what-to-say</i>	¿el sujeto piensa lo que va a decir?
<i>gives-encouragement</i>	¿el sujeto alienta a su compañero?
<i>difficult-for-partner-to-speak</i>	¿el sujeto le hace difícil hablar a su compañero?
<i>bored-with-game</i>	¿el sujeto está aburrido con el juego?
<i>dislikes-partner</i>	¿al sujeto no le agrada su compañero?

Cuadro 1. Preguntas sobre las percepciones sociales realizadas a los anotadores

Variable	Descripción
<i>F0 Mean</i>	Valor medio de la frecuencia fundamental
<i>F0 Max</i>	Valor máximo de la frecuencia fundamental
<i>Int Mean</i>	Valor medio de la intensidad
<i>Int Max</i>	Valor máximo de la intensidad
<i>NHR</i>	Noise-to-harmonics ratio
<i>Shimmer</i>	Shimmer medido
<i>Jitter</i>	Jitter medido
<i>Sílabas/seg</i>	Cantidad de sílabas por segundo
<i>Fonemas/seg</i>	Cantidad de fonemas por segundo

Cuadro 2. Variables acústico-prosódicas medidas

- *f0* refiere a la frecuencia fundamental de una onda, que es el recíproco del período de ésta. El *tono* o *pitch* es la percepción que tenemos de la frecuencia fundamental, que nos marca cuán agudos o graves son los sonidos.
- *Intensity* refiere al volumen o intensidad de la onda. Ésta mide la amplitud de la onda, y es la percepción de cuán fuerte es el sonido.
- *jitter* y *shimmer* se refieren, en un intervalo de tiempo, a los desplazamientos de la onda de la verdadera periodicidad y de la amplitud, respectivamente. Están asociadas con la percepción de la calidad de la voz.
- Un *fonema* es la articulación simple de sonidos del habla, tanto de vocales como de consonantes. Ejemplos de fonemas son los sonidos de las letras u, a, s, k en español.
- El *noise-to-harmonics ratio* (abreviado NHR) puede considerarse como una medida de calidad de la voz, que cuantifica la proporción de ruido que hay en ésta.

2.2. Descripción del método TAMA

En el trabajo de Kousidis [13] se introdujo un método novedoso para el análisis del *entrainment* acústico-prosódico. Esta técnica consiste en armar dos series de tiempo para cada uno de los interlocutores y luego utilizar herramientas de análisis bivariado sobre las series construídas.

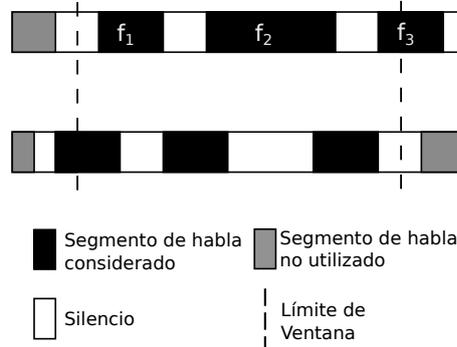


Figura 2. Gráfico de la separación del diálogo en ventanas. Fuente [12]

Para construir la serie de tiempo de cada hablante debemos, en primer lugar, dividir el diálogo en ventanas solapadas de igual tamaño. A la diferencia entre ventana y ventana llamaremos *frame step*, y al tamaño de ventana *frame length*. Consideraremos sólo los segmentos de habla que tengan intersección con la ventana dentro de cada ventana. En la Figura 2 se ilustra el proceso: las líneas punteadas marcan los límites de la ventana, los intervalos coloreados en negro los segmentos de habla considerados, y en gris los segmentos no considerados.

Luego se calculan los valores de la serie de tiempo para cada hablante y cada variable acústico-prosódica mediante la siguiente fórmula:

$$\mu = \sum_{i=1}^N f_i d'_i \quad (1)$$

donde i itera sobre las elocuciones dentro del *frame*, d'_i es la duración relativa del segmento (respecto del tiempo total hablado en toda la ventana) y f_i es el valor de la *variable acústico-prosódica* que estamos midiendo.

Como se ve en la ecuación 1, el valor que calculamos es una media ponderada del valor de la variable por la duración de las elocuciones. Así, por ejemplo, al calcular una serie de tiempo sobre la intensidad, la contribución de interjecciones (*ah!* por ejemplo), que suelen tener altos valores, estará atenuada por sus breves duraciones.

Una vez obtenidas las series de tiempo respectivas, una posible medida del *entrainment* se puede obtener midiendo cuánto influye una serie sobre otra, considerándolas a ambas como parte de un sistema donde ambas interactúan. Este *entrainment*, entonces, sería direccional: queremos medir cuánto influye el interlocutor *A* sobre el interlocutor *B* y viceversa. Puede darse el caso en que ambos tengan fuerte interacción, en tal caso hablamos de *feedback*.

Para medir cuánto se mimetizan las dos series, el trabajo de Kousidis y otros usa la función de correlación cruzada (f.c.c) [5], que mide cuánto se parecen la

serie X e Y aplicando un desplazamiento k , lo cual nos arroja como resultado un valor entre -1 y 1 (similar al coeficiente de correlación de la estadística clásica). Podemos aproximar la f.c.c mediante la fórmula de la correlación cruzada muestral

$$r_{AB}(k) = \begin{cases} \frac{\sum_{t=k+1}^n (A_t - \mu_A)(B_{t-k} - \mu_B)}{\sqrt{\sum_{t=1}^n (A_t - \mu_A)^2 \sum_{t=1}^n (B_t - \mu_B)^2}} & \text{si } k \geq 0 \\ \frac{\sum_{t=-k+1}^n (B_t - \mu_B)(A_{t+k} - \mu_A)}{\sqrt{\sum_{t=1}^n (A_t - \mu_A)^2 \sum_{t=1}^n (B_t - \mu_B)^2}} & \text{si } k < 0 \end{cases} \quad (2)$$

Podemos ver que, si $k \geq 0$, lo que hacemos es, a grandes rasgos, calcular la correlación de Pearson entre A y B , pero tomando los $n - k$ últimos valores de A y los $n - k$ primeros de B . Si $k < 0$, lo hacemos entre A y B , pero desplazando en sentidos inversos. Viéndolo de otra forma, si $k \geq 0$, estamos midiendo cuánto influye B sobre A contemplando un desplazamiento de k puntos; si $k \leq 0$ medimos la influencia de A sobre B a misma distancia. La utilización de estos desplazamientos está explicada en el trabajo de Gravano et al [9], donde se menciona que la influencia de los hablantes no es necesariamente inmediata sino que puede tener algunos segundos de demora para tomar lugar.

Para cada conversación, se estima entonces el correlograma cruzado, considerando desplazamientos tanto positivos como negativos. Hecho esto, en el mencionado trabajo [12] sólo analizan la significancia de los resultados de la correlación cruzada, enumerando aquellos lags en los cuales esto ocurrió.

2.3. TAMA sobre Columbia Games

En el trabajo de Kousidis [12, 13] se discute la disyuntiva de elegir un tamaño de ventana y step para el método: ventanas demasiado chicas pueden causar que

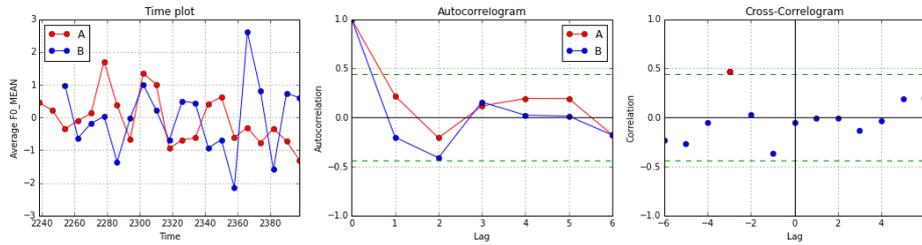


Figura 3. Time-plot producido por TAMA, junto a su autocorrelación y correlación cruzada

no hayan segmentos de habla en ellas, mientras que un tamaño de ventana demasiado grande suavizaría en exceso la serie de tiempo. En el trabajo de De Looze et al [7] donde se refina la técnica se mencionan dos posibles soluciones para el problema de los puntos faltantes: interpolar o repetir el punto anterior de la serie. Estos enfoques, sin embargo, pueden dar lugar a valores de *entrainment* artificialmente altos por la construcción misma de la serie, ya que nos generaría puntos correlacionados fuertemente entre sí en cada una de las series de los hablantes. Por otro lado, descartar aquellas conversaciones que tengan puntos faltantes puede ser demasiado restrictivo y eliminar de nuestro corpus una gran cantidad de datos valiosos. Teniendo estas cosas en mente, decidimos aceptar series de tiempo con datos faltantes, que pueden ser producto de ventanas sin segmentos de habla o con algunos demasiado pequeños que imposibiliten la medición de las variables acústico-prosódicas.

A su vez, se modificó el tamaño de la ventana para ajustarlo a nuestro corpus. En vez del step de 10" y tamaño de 20", optamos por 8 y 16 segundos respectivamente luego de efectuar un análisis con el fin de encontrar un balance entre el tamaño de la ventana y la cantidad de indefiniciones. Considerando estos parámetros, se procedió a calcular las dos series de tiempo para cada conversación y cada variable acústico-prosódica. De estas tareas, sólo nos quedamos con aquellas que tengan al menos 5 puntos definidos para cada serie, de manera que tenga sentido poder calcular la correlación cruzada más adelante. Con esto, no sólo nos interesa la duración de la charla, sino cierta calidad de las series generadas.

2.4. Métricas de entrainment

A diferencia del trabajo de Kousidis [12] en el cual sólo se considera la existencia de entrainment o no mediante la significancia de la correlación cruzada muestral, en nuestro trabajo definimos dos métricas basadas en esta función que finalmente utilizamos en un análisis de regresión con las variables sociales del corpus.

En primer lugar, definimos $\mathcal{E}_{AB}^{(1)}$ como el valor de $r_{AB}(k)$ con mayor valor absoluto, dado $k \leq 0$. Análogamente lo definimos para $\mathcal{E}_{BA}^{(1)}$, usando $k \geq 0$. Esta métrica es esencialmente la definida en el trabajo de Kousidis. En segundo lugar, definimos una segunda métrica $\mathcal{E}_{XY}^{(2)}$ como el valor absoluto de la primera, es decir $\mathcal{E}_{XY}^{(2)} = |\mathcal{E}_{XY}^{(1)}|$, donde XY puede ser AB o BA .

Vale remarcar que la segunda métrica considera de forma positiva la asincronía entre las series de tiempo, también conocido en la literatura como antimimicry o *disentrainment* [4]. Este fenómeno refiere al proceso por el cual uno de los hablantes no imita al otro sino más bien todo lo contrario, acentúa alguna diferencia. Si bien estudios de larga data [1, 6] lo emparentan con una connotación negativa, otros trabajos más recientes [11, 14] sugieren que puede entenderse este fenómeno como una conducta de adaptación cooperativa. No sólo eso, sino que este fenómeno de mimetización complementaria podría ser incluso más prevalente que la mimetización a secas [14].

	<i>Int Max</i>	<i>Int Mean</i>	<i>F0 Mean</i>	<i>F0 Max</i>	NHR	Fon/seg	Sil/seg	SHIMMER	JITTER
contributes		+	+++	+	++				
clear	+++		+		+++		+		
engaged		+	+++						+
planning									
encourages								+	
difficult	--	++				-			
bored			---		+				
dislikes									

Cuadro 3. Resultados de la regresión de efectos fijos. El símbolo + representa valor significativo y positivo de la pendiente de la regresión de efectos fijos, mientras que - representa significativo y negativo. + representa $p < 0,10$, ++ $p < 0,05$, y +++ $p < 0,01$. Análogamente para -, --, y ---

2.5. Análisis de regresión

Luego de construir las series de tiempo para cada una de las conversaciones (y cada una de las variables acústico-prosódicas) que seleccionamos anteriormente, nos interesó evaluar la relación entre los valores de las métricas de *entrainment* y las distintas variables sociales. Con esto en mente, planteamos un modelo de regresión lineal tomando como nuestra variable *explicativa* la mimetización, y la variable *dependiente* será la variable social elegida. Este análisis de regresión nos permitió observar cuál es la variación conjunta de ellas.

Nuestra hipótesis consistió en que la mimetización (por ejemplo, en la intensidad o pitch) se relacionaría de manera directa con ciertas variables sociales de connotación positiva (por ejemplo, la compenetración en el juego) y que se relacionaría de manera inversa con aquellas de carácter negativo (el aburrimiento o el desagrado por su compañero), siguiendo la línea de trabajos previos [9].

Utilizamos *regresión de efectos fijos* [10, chap 16], un modelo que ayuda a controlar la heterogeneidad no observada cuando ésta es constante en el tiempo para cada sujeto del sistema. Asumimos que estos factores son inherentes a la conversación entre el hablante y su interlocutor, y por este motivo, definimos los sujetos como cada uno de los hablantes y sus respectivas sesiones. No nos importa si el mismo sujeto se repite en otra sesión: cada hablante de una sesión es un sujeto distinto para el modelo de efectos fijos.

3. Resultados

La primer métrica que definimos, $\mathcal{E}_{AB}^{(1)}$, no obtuvo resultados significativos para las pendientes, por lo cual las omitimos del presente análisis. Por otro lado, la métrica $\mathcal{E}_{AB}^{(2)}$ dio valores sustancialmente apreciables. La Tabla 3 presenta una versión resumida de los resultados para la regresión de las variables sociales sobre esta métrica, para cada una de las variables acústico-prosódicas, marcando con + (aproximadamente significativo), ++ si $p < 0,05$ y +++ si $p < 0,01$.

Podemos observar que para las primeras cinco variables acústico-prosódicas, y las tres primeras variables sociales tenemos un conjunto de pendientes positivas

significativas (o aproximadamente). Estas variables (*contributes-to-completion*, *making-self-clear*, y *engaged-with-game*) representan percepciones positivas, que era lo que esperábamos ver: a mayor valor de *entrainment*, mayor valor de estas variables sociales positivas. Esta tendencia es principalmente notoria para las variables acústico-prosódicas relacionadas al tono y a la intensidad. Para aquellas variables sociales de connotación negativa también se encontraron pendientes menores a cero, aunque en este caso tan sólo dos son significativas.

A su vez, puede destacarse que esta medida del *entrainment* es consistente con otras métricas definidas en otros trabajos, como las construidas en [9] sobre anotaciones discretas de los patrones entonacionales usando la convención ToBI [17].

4. Conclusiones y trabajo futuro

En el presente trabajo, analizamos cómo dos métricas dinámicas del fenómeno conocido como *entrainment* en el plano acústico-prosódico se relacionan con las percepciones de aspectos sociales de la interacción entre los participantes. Ambas métricas pueden computarse en forma automática a partir de las grabaciones de las conversaciones y con transcripciones alineadas temporalmente al audio. Todo este análisis se da en el contexto de un juego orientado a tareas, que comprende interacciones de una naturaleza muy similar a las de una interfaz humano-computadora.

Estas métricas fueron construidas a través del análisis de series de tiempo, y apuntan a cuantificar cuánto se imitan o mimetizan los hablantes en términos de sus variables acústico-prosódicas. En primer lugar, contemplamos una métrica que penaliza el *disentrainment* con valores negativos. Se aplicó análisis de regresión sobre esta métrica, y los resultados que dio no fueron significativos. En segundo lugar, construimos una métrica que valora de igual manera el *entrainment* y el *disentrainment*, de acuerdo a trabajos previos que sugerían que el segundo fenómeno puede considerarse en algunas circunstancias como un mecanismo de adaptación cooperativa. Al efectuar el análisis de regresión sobre esta métrica, los resultados fueron significativos y consistentes con la hipótesis planteada de que el *entrainment* se relaciona positivamente con características sociales favorables de la conversación, mientras que lo hace de manera inversa con aquellas negativas. A su vez, esta métrica usada en esta tesis se comporta de manera consistente preservando las relaciones expuestas en otros trabajos entre el *entrainment* y aquellas variables sociales de carácter positivo y negativo.

Una contribución importante de este trabajo es la validación de la métrica introducida por Kousidis et al [12], dando indicios de que ésta efectivamente captura rasgos relevantes de la interacción, como por ejemplo la percepción social de la conversación. Igual de importante es el uso del valor absoluto de la correlación cruzada (introducido en el presente trabajo) como medida unificadora del *entrainment* y *disentrainment* y que remarca la importancia del segundo fenómeno dentro de la comunicación verbal, a la luz de últimos trabajos acerca de la divergencia en el diálogo.

A pesar de que los resultados son prometedores, siguen siendo preliminares y su robustez requiere de más validaciones. Como trabajo futuro, proponemos reproducir estos análisis sobre otros corpus de habla, como por ejemplo Switchboard². Otra dirección posible es utilizar herramientas de análisis multivariado de series de tiempo sobre las diferentes variables acústico-prosódicas y sobre la base de esto construir nuevas métricas del *entrainment* prosódico.

Referencias

1. R. Y. Bourhis, H. Giles, and H. Tajfel. Language as a determinant of welsh identity. *European Journal of Social Psychology*, 3(4):447–460, 1973.
2. S. E. Brennan. Lexical entrainment in spontaneous dialog. *Proceedings of ISSD*, 96:41–44, 1996.
3. J. K. Burgoon, L. A. Stern, and L. Dillman. Interpersonal adaptation: Dyadic interaction patterns. 1995.
4. T. L. Chartrand and J. A. Bargh. The chameleon effect: the perception–behavior link and social interaction. *Journal of personality and social psychology*, 76(6):893, 1999.
5. C. Chatfield. *The analysis of time series: an introduction*. CRC press, 2013.
6. J. M. Dabbs Jr. Similarity of gestures and interpersonal influence. In *Proceedings of the annual convention of the American Psychological Association*. American Psychological Association, 1969.
7. C. De Looze, S. Scherer, B. Vaughan, and N. Campbell. Investigating automatic measurements of prosodic accommodation and its dynamics in social interaction. *Speech Communication*, 58:11–34, 2014.
8. A. Gravano. *Turn-taking and affirmative cue words in task-oriented dialogue*. Columbia University, 2009.
9. A. Gravano, Š. Benuš, R. Levitan, and J. Hirschberg. Backward mimicry and forward influence in prosodic contour choice in standard american english. In *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.
10. D. N. Gujarati and D. C. Porter. Essentials of econometrics. 1999.
11. P. G. Healey, M. Purver, and C. Howes. Divergence in dialogue. *PloS one*, 9(6):e98598, 2014.
12. S. Kousidis, D. Dorran, C. McDonnell, and E. Coyle. Times series analysis of acoustic feature convergence in human dialogues. In *Proceedings of Interspeech*, 2008.
13. S. Kousidis, D. Dorran, Y. Wang, B. Vaughan, C. Cullen, D. Campbell, C. McDonnell, and E. Coyle. Towards measuring continuous acoustic feature convergence in unconstrained spoken dialogues. 2008.
14. R. Levitan, Š. Benuš, A. Gravano, and J. Hirschberg. Acoustic-prosodic entrainment in slovak, spanish, english and chinese: A cross-linguistic comparison. In *16th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, page 325, 2015.
15. R. Levitan and J. B. Hirschberg. Measuring acoustic-prosodic entrainment with respect to multiple levels and dimensions. 2011.

² <https://catalog.ldc.upenn.edu/LDC97S62>

16. R. Pieraccini and J. Huerta. Where do we go from here? research and commercial spoken dialog systems. In *6th SIGdial Workshop on Discourse and Dialogue*, 2005.
17. J. F. Pitrelli, M. E. Beckman, and J. Hirschberg. Evaluation of prosodic transcription labeling reliability in the tobi framework. In *ICSLP*, 1994.
18. D. Reitter, F. Keller, and J. D. Moore. Computational modelling of structural priming in dialogue. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 121–124. Association for Computational Linguistics, 2006.
19. N. G. Ward, A. G. Rivera, K. Ward, and D. G. Novick. Root causes of lost time and user stress in a simple dialog system. In *Ninth European Conference on Speech Communication and Technology*, 2005.

Un asistente flexible para la formación de equipos en base a técnicas SAT

Ignacio Gatti

ISISTAN (CONICET-UNCPBA)
ignacio.gatti@isistan.edu.ar

Resumen La formación de equipos es clave a la hora de llevar adelante un proyecto, porque las características de cada uno de los miembros influyen en el desempeño grupal. En este artículo, se propone un enfoque automatizado que modela la formación de equipos como un problema de satisfacción de restricciones ponderadas y se utiliza un motor de optimización pseudo-booleano para su resolución. El asistente propuesto considera cuatro dimensiones de los usuarios: estilos psicológicos, roles de equipo, redes sociales y competencias técnicas. Un aspecto central del enfoque radica en la flexibilidad que ofrece al “seleccionador” para definir sus propios modelos de equipo en términos de restricciones y preferencias. Para analizar nuestro enfoque, se estudió, por un lado, su comportamiento ante un asistente que propone una visión restrictiva del modelado de equipos, pero que utiliza un enfoque de resolución similar al adoptado aquí; y por otro lado, se analizó la calidad de las soluciones para instancias del problema donde los modelos de equipos se abren a otras posibilidades.

1. Introducción

La formación de equipos es una tarea que se presenta siempre problemática y compleja. A la hora de comenzar un proyecto, todo gerente -o quien juegue el rol de “seleccionador” de recursos humanos- debe decidir y escoger un conjunto de personas, de manera tal que ese grupo asegure un desempeño exitoso. Sin embargo, dicha elección no siempre es una tarea fácil, con lo cual mientras muchos equipos alcanzan los objetivos propuestos, otros tantos fracasan.

Según varios autores (por ej., Belbin [1][2]; Mumma [3]; Myers [4][5]; Mitchell, Nicholas y Boyle [6]; Pieterse, Kourie y Sonnekus [7]; Winter [8]), el problema principal radica en la noción que se tiene de equipo y en los criterios adoptados para llevar a cabo la selección. El no tener una definición clara acerca de lo que es un equipo trae aparejado que los miembros se elijan casi al azar, utilizando criterios dudosos o algunas veces contradictorios.

Para Belbin [1][2], un equipo es “una congregación de personas donde cada uno de ellos desempeña un rol que es comprendido por el resto de los miembros. Éstos negocian entre sí el reparto de roles y desempeñan de manera más eficaz aquellos que les son más naturales”. A partir de esta definición, se puede encontrar un abanico de estudios y teorías que buscan establecer leyes y principios

acerca de qué características debe tener un conjunto de personas para poder ser un equipo. A grandes rasgos, estos aspectos se pueden dividir en dos categorías: psicológicas-sociales y técnicas. Mientras que las primeras están relacionadas con el perfil psicológico de cada integrante y cómo se desenvuelve éste dentro de un grupo; las segundas, se vinculan con las capacidades técnicas propias de la tarea concreta a desarrollar.

En líneas generales, se puede describir el proceso de formación como la selección de un subconjunto de personas con perfiles específicos. Se puede observar que dicha tarea encierra una doble dificultad: la elección en sí misma, y la ponderación de todos los potenciales equipos. Particularmente, desde el área de Inteligencia Artificial se han realizado diversas investigaciones que proponen técnicas para la automatización de dicho proceso. Entre ellos, el trabajo propuesto en [9] aborda esta problemática como un problema de satisfacción de restricciones, donde se presupone que existe una configuración fija de restricciones para alcanzar un equipo óptimo. Dicho trabajo marca el punto de partida del presente, dado que no siempre es posible considerar la configuración de restricciones como algo fijo, pues depende del punto de vista del seleccionador.

En este contexto, el presente trabajo propone un asistente para la formación de equipos de trabajo utilizando un enfoque de optimización de restricciones pseudo-booleano. Se considera que un equipo es atravesado por cuatro características principales: el perfil psicológico de sus miembros, según la teoría de Myers-Briggs [5]; los roles de equipo que se deben cubrir, según la concepción de Mumma [3]; las redes sociales, siguiendo las ideas de Anderson [10]; y las competencias técnicas que exige el proyecto a desarrollar (por ej., conocimientos de Java), entendiendo que el equipo se conforma para la resolución de una situación particular y concreta. Sin embargo, a diferencia de [9], en este enfoque no se busca un único equipo “ideal”, sino que permite que el seleccionador pueda definir y experimentar con distintos modelos, que luego son resueltos por un componente de optimización.

El resto del trabajo se organiza de la siguiente manera. La sección 2 se aboca a la descripción del trabajo propuesto para la problemática de la formación de grupos. La sección 3 plantea cómo modelar la problemática con un enfoque de optimización de restricciones pseudo-booleana. La sección 4 muestra y discute los resultados experimentales obtenidos. Por último, la sección 5 presenta las conclusiones y trabajos futuros.

2. Enfoque propuesto

En este trabajo se presenta un asistente automatizado, que utiliza un enfoque de optimización de modelos pseudo-booleanos para generar potenciales equipos en base a las preferencias de un seleccionador. En líneas generales, se busca que el asistente ofrezca libertad al seleccionador para que éste pueda proponer sus propios modelos. Por ejemplo, el seleccionador podría considerar que para un grupo es importante y necesario que todos los roles (propuestos por Mumma)

estén cubiertos y, a su vez, que es deseable pero no determinante que cada rol de cada integrante se corresponda con sus preferencias personales y, finalmente, que es bueno que los integrantes tengan temas de interés común. Para otro grupo, el seleccionador podría establecer como excluyente que estuviesen cubiertos todos los roles y que todos los integrantes del equipo tuviesen un vínculo.

Tal como muestra la Figura 1, para poder generar equipos que satisfagan los requerimientos del administrador (o seleccionador), es necesario, por un lado, que las teorías psicológicas y sociales que dan soportes a la formación de equipos sean modeladas como restricciones según el enfoque pseudo-booleano; y por el otro, que los perfiles de los usuarios estén cargados en el sistema. De esta manera, el asistente toma el modelo de equipo definido, y lo mapea a un sistema de desigualdades pseudo-booleana que es resuelto utilizando el framework SAT4J¹. Desde el punto de vista teórico, se escogieron un conjunto de referentes que fue-

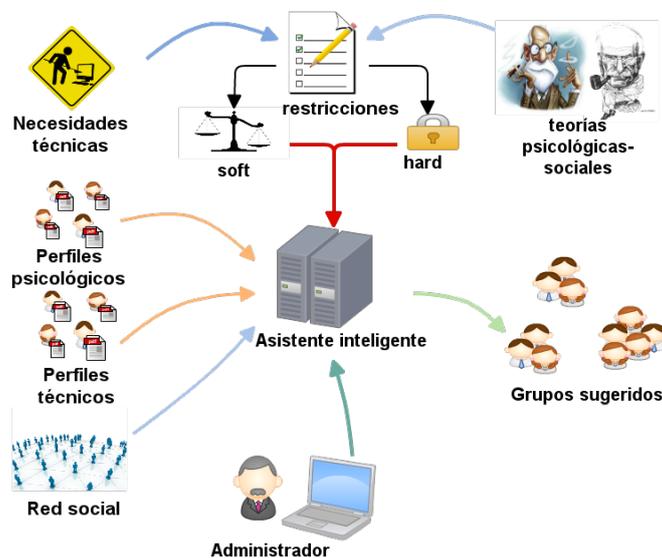


Figura 1. Visión general del asistente inteligente

ron las bases para definir conceptualmente un equipo y las características que lo distingue de un grupo. Siguiendo las ideas propuestas principalmente en el trabajo [9], para la definición del aspecto psicológico-social se tomaron: los estudios de perfiles psicológicos realizados por Myers-Briggs; la teoría de Roles propuesta por Mumma; los estudios realizados por Anderson sobre las redes sociales. Myers-Briggs [5] propone un modelo para clasificar las personalidades, derivado de la teoría de Tipos Psicológicos de Carl Jung; puntualmente, en sus estudios

¹ <http://www.sat4j.org/>

se diferencian cuatro dimensiones -la fuente de motivación, el modo de percibir la información, el modo de utilizar la información, la postura frente al mundo-, cada una con dos funciones que permiten a las personas desenvolverse de determinada manera en su vida. Según esta línea psicológica, cada individuo, de acuerdo a su personalidad, se inclina por algunas funciones. Asimismo, el trabajo [5] analiza la influencia de las personalidades dentro de un grupo de trabajo.

Por el lado de la teoría de Roles de Mumma [3], siguiendo la misma concepción de equipo que Belbin, establece ocho roles que son esenciales para el éxito de un equipo a la hora de afrontar un proyecto: Líder, Moderador, Creador, Innovador, Manager, Organizador, Evaluador y Finalizador. Además, entiende que cada proyecto tiene un ciclo de vida, dividido en cuatro fases donde, en cada uno, dos roles son indispensables para su concreción, ofreciendo una visión integral entre equipo y tarea a desarrollar.

El enfoque de redes sociales propuesto por Anderson [10] analiza la estructura de relaciones sociales subyacente que se da en todo grupo social. Específicamente, encuentra una fuerte conexión entre las estructuras sociales y las decisiones que toma cada individuo a la hora de elegir un grupo de trabajo. De éstos deriva que aquellos equipos donde los integrantes comparten más intereses -es decir, que hay una red social densamente conectada- resultan ser más eficiente que aquellos cuya red social es mala.

Finalmente, la definición de las capacidades técnicas se basa en la elección de un conjunto de requisitos que suelen buscarse en los desarrolladores de software (conocimiento de programación, manejo de base de datos, dominio del inglés), entendiéndose que cada proyecto exige la puesta en acción de habilidades y conocimientos puntuales para alcanzar su desarrollo exitoso.

3. La formación de equipos bajo un enfoque SAT

Tal como se describe en [9], la asignación de personas a un equipo, donde la elección radica tanto en las características del individuo, así como en los aspectos a cubrir dentro del grupo, puede ser visto como un problema de satisfacción de restricciones. Básicamente, siguiendo las ideas de Petke [11], estos tipos de problemas se definen formalmente como una tripleta $\langle V, D, C \rangle$, donde V es el conjunto de variables, D es dominio de los valores que pueden adoptar las variables, y C es el conjunto de restricciones.

Al instanciarlo en esta problemática en particular, D es un conjunto donde existen dos valores posibles 0, 1, que se traducen a: forma parte del equipo (1) o no (0). Por el lado de las restricciones, es necesario observar que en el “mundo real”, a la hora de decidir, éstas enfatizan distintos aspectos: mientras hay algunas restricciones que son obligatorias y se deben cumplir para que las soluciones sean válidas; otras, sólo afectan la calidad de la solución, es decir, pueden existir soluciones que las violen (a costa generalmente de una calidad baja). Las primeras se denominan *restricciones fuertes* y las segundas *restricciones débiles*. Sin embargo, la definición de una restricción como *fuerte* o *débil* no es algo rígido,

sino que, en muchas circunstancias, depende de quién esté formando los equipos. La introducción de diferentes tipos de restricciones nos lleva a redefinir el problema, ahora como un problema de optimización. En otras palabras, las *restricciones débiles* implican que las soluciones propuestas, además de cumplir con todas las *restricciones fuertes*, tendrán asociadas una penalización que se desprende del grado de violación de las primeras. De esta manera, se puede definir formalmente a esta problemática como $F(A) = \sum C_i(A) * cost_i(A)$, donde A es un vector de asignación –por ejemplo, $A = (x_1 = 0, \dots, x_i = 1, \dots, x_n = 1)$ –, y cada restricción tiene un costo asociado – en el caso de las *restricciones fuertes* el costo es infinito (∞) –.

Desde el área de Inteligencia Artificial es posible encontrar diversos enfoques y técnicas, tales como backtracking, búsqueda local, algoritmos evolutivos, que permiten abordar un problema de optimización como el propuesto en el párrafo anterior. En este trabajo, se exploró un enfoque de optimización de modelos pseudo-booleanos, porque permitía la posibilidad de definir modelos donde se conjugaban de manera natural los dos tipos de restricciones. En este trabajo, la noción de expresión pseudo-booleana es entendida como:

$$F(x_1, \dots, x_n) = \sum c_i * x_i + \sum c_{i,j} * x_i * x_j + \dots \quad c_i, c_{i,j} \in \mathbf{R} | x_i \in B_2 \quad (1)$$

A partir de dicha definición, este enfoque propone definir un sistema de inecuaciones con una función objetivo a optimizar. Para resolver los modelos pseudo-booleanos se utilizó el framework SAT4J², que ofrece un abanico de algoritmos para manipular estos tipos de sistemas.

Desde el punto de vista del modelado de las restricciones, a diferencia de otros trabajos, el asistente posee la flexibilidad para que el seleccionador pueda decidir qué restricciones se deben tener en cuenta para cada equipo y establecer de qué tipo son. Para ello, fue necesario modelar cada restricción tanto *fuerte* como *débil*. Por ejemplo, según la teoría de Roles de Mumma, para que un grupo sea un equipo se deben dar dos condiciones: i) que todos los roles deben estar cubiertos; y ii) se debe evitar la “sobrepoblación” de un rol dentro del equipo. Con lo cual, si se piensa dichas premisas como condiciones indispensables (*restricciones fuertes*), por cada rol se puede definir las siguientes inecuaciones:

$$\begin{aligned} x_i + x_j + \dots + x_r &\geq 1 \\ x_i + x_j + \dots + x_r &\leq m/2 \end{aligned} \quad (2)$$

donde x_i, x_j, x_r son personas que desempeñan el mismo rol

Mientras que, si se las considera como aspectos deseables en el equipo a formar (*restricciones débiles*), por cada rol se puede derivar las siguientes expresiones:

² <http://www.sat4j.org/>

$$\begin{aligned}
& \text{mín } C_R * x_R \\
& x_i + x_j + \dots + x_r \geq 1 \\
& -Cota * x_R + x_i + x_j + \dots + x_r \leq m/2
\end{aligned} \tag{3}$$

donde x_i, x_j, x_r son personas que desempeñan el mismo rol, x_R es la variable de contrapeso, $Cota$ es un valor tan alto que haga verdadera la inecuación para el peor caso, y C_R es el costo por relajar la condición

Asimismo, el resto de las reglas derivadas de las teorías escogidas para la definición del concepto de equipo fueron modeladas de forma análoga al ejemplo anterior. Básicamente, y siguiendo el análisis realizado en [9], de las teorías de Mumma, Myers-Briggs y Anderson, así como de las competencias técnicas, se obtuvieron una serie de condiciones que posteriormente fueron mapeadas a un modelo matemático pseudo-booleano. Y es aquí donde radica el punto neurálgico del trabajo, porque en ese mapeo se desliga a cada regla de una representación estática, establecida "arbitrariamente" por el programador. En cambio, al definirla tanto como posibilidad (*restricción débil*) o como condición necesaria (*restricción fuerte*), el modelo se completa a partir de las necesidades de cada seleccionador particular, ofreciendo la posibilidad de que cada restricción se pueda mover libremente entre lo obligatorio y lo deseable.

Puntualmente, para crear un modelo en el asistente, tal como muestra la Figura 2, el seleccionador debe definir cuántos equipos desea formar. A partir de allí, el asistente genera una pestaña por cada equipo, donde se puede elegir y configurar cada una de las restricciones modeladas. Por ejemplo, si el seleccionador desea formar dos equipos, éste tiene la posibilidad de configurar en el equipo 1 la restricción *Roles Balanceados* como *débil* y ajustar el costo que tiene asociado la violación de dicha condición; mientras que, para el equipo 2 puede definir esta misma condición como *fuerte*. De esta manera, cada condición es independiente para cada uno de los equipos que se desean formar en el modelo; es decir, retomando las palabras del párrafo anterior, cada restricción se puede mover entre lo obligatorio y lo deseable.

4. Resultados experimentales

Desde el punto de vista experimental, se llevaron a cabo dos tipos de pruebas: una comparativa contra el asistente desarrollado en [9] y otra contra sí mismo. En la primera prueba se estudió el comportamiento de ambos asistentes ante un mismo caso, de manera tal de poder comparar el rendimiento de ambos -desde el punto de vista del tiempo de ejecución. El objetivo de la segunda prueba fue analizar cómo variaba el comportamiento del asistente a medida que se definían instancias más complejas y con modelos de restricciones más relajadas, tomando como referencias el tiempo de ejecución y la calidad de la solución obtenida.

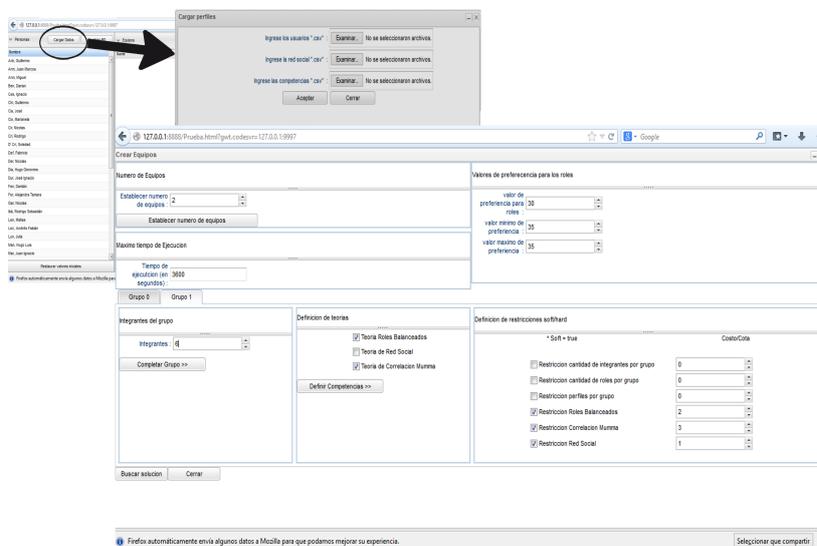


Figura 2. Interfaz del asistente. Pantalla para la definición del modelo

Al momento de medir y comparar la calidad de los equipos, es decir, de determinar cuanto se acercaban al modelo ideal de equipo definido por el seleccionador, se decidió tomar el mismo indicador que en el trabajo [9]. Básicamente, a cada equipo se le asignó una penalización que se correspondía con las *restricciones débiles* violadas. Sin embargo, a simple vista se observa que existen diversos grados de violación: no es lo mismo tener un solo rol de Mumma desbalanceado con un integrante de más a tener todas las categorías desequilibradas. Por eso, el indicador utilizado para la penalización, crece, por un lado, de acuerdo al grado de incumplimiento. Por otro lado, también se tuvo en cuenta que, para el seleccionador, podría ocurrir que no todas las *restricciones débiles* tuviesen el mismo porcentaje de influencia; con lo cual, a cada una de las teorías propuestas -Mumma, Myers-Briggs, Anderson y competencias técnicas- se les asignó un peso (un valor entre 0 y 1), que además podía ser modificado por el seleccionador, con el fin de poder plasmar esas jerarquías. Formalmente, dicho indicador se expresa algebraicamente de la siguiente manera:

$$P(x_i) = \frac{C_{Mumma} * \sum x_i + C_{Myers-Briggs} * \sum x_j + C_{Anderson} * \sum x_k}{factor_n} \quad (4)$$

donde x_i, x_j, x_r son personas que violan alguna de las teorías definidas en el modelo; $C_{Mumma}, C_{Myer-Briggs}, C_{Anderson}$ son los pesos asociados a las reglas derivadas de cada teoría; y $factor_n$ es un factor de normalización.

Asimismo, para calcular la penalización global de una solución (es decir, un conjunto de grupos), se utilizó la media aritmética, por ser un indicador sencillo que ofrece una rápida aproximación a la calidad de los equipos formados.

De esta manera, resulta claro el criterio adoptado para la comparación: cuanto más se acerca a 0 la penalización de un equipo, más próximo se encuentra del equipo ideal pedido por el seleccionador. Vale la pena destacar que dicho criterio, define la calidad a partir de su “negatividad”: básicamente, lo que se observa en un equipo son los atributos que le faltan; aquellas reglas que no cumple. Con lo cual, en los experimentos realizados el foco se colocó en analizar cuánto se “desviaban” éstos de las restricciones definidas; dejando de lado, otros aspectos que se podrían llamar “positividades”, donde se buscaría analizar, por ejemplo, el impacto que produciría poseer un equipo con determinado líder, tratando de comprender cómo impacta éste líder y no otro en ese equipo particular.

Para la realización de los experimentos, se tomó un dataset con 44 estudiantes, que habían realizado el curso de “Trabajo Colaborativo Soportado por Computadoras” en el año 2011 en la carrera de Ingeniería de Sistemas de la UNCPBA. Asimismo, las pruebas fueron ejecutadas sobre un procesador Intel Pentium B960 2.20 GHz con 4 GB de memoria RAM, y Sistema Operativo Windows 64 bits. Para la primera prueba, los casos de estudios contaron con la siguiente configuración: las restricciones derivadas de las teorías de *Mumma* y *Myers-Briggs* se definieron como *fuertes*, mientras que las restricciones de *Roles Balanceados*, *Correlación entre roles y perfiles psicológicos* y *Red Social* como *débiles*, que es la configuración utilizada en [9]. A partir de ésta, se solicitó a ambos asistentes un conjunto de 10 soluciones posibles para la formación de 2 y 3 equipos, donde el mínimo de integrantes fue 4 y el máximo, 10. La elección se vio influenciada principalmente por las características del conjunto de datos utilizado. Al realizar una clasificación de ellos de acuerdo a los roles y perfiles, se observó que mientras que para algunos segmentos había superpoblación, en otros había escasez. Con lo cual, al no haber una distribución homogénea, la posibilidad de formar varios equipos y que todos cumplieren con el modelo definido previamente, quedaba atada a grupos minoritarios.

Desde el punto de vista de los tiempos de ejecución, se observó un rendimiento similar entre ambos, incluso nuestro enfoque resultó más lento en algunos casos. Básicamente, esto se debió a la lógica que proponen las interfaces de los algoritmos de SAT4J: al solicitar más de una solución, se obliga a éste a volver a recorrer, desde el inicio, el espacio de soluciones -una vez por cada nueva solución-. Sin embargo, lo más interesante fue la diferencia en los tiempos de ejecución entre escenarios pares e impares. Entre las causas de esta particularidad, la más plausible fue la complejidad que adoptaron las restricciones que debían considerar que la mitad de los integrantes poseyeran una característica determinada y la otra mitad una diferente, porque mientras que para las pares la división resultaba exacta, en el caso de las impares, uno de los conjuntos debía tener inexorablemente un individuo más que el otro, abriendo más posibilidades para analizar. Esta peculiaridad que trae aparejada los equipos impares se de-

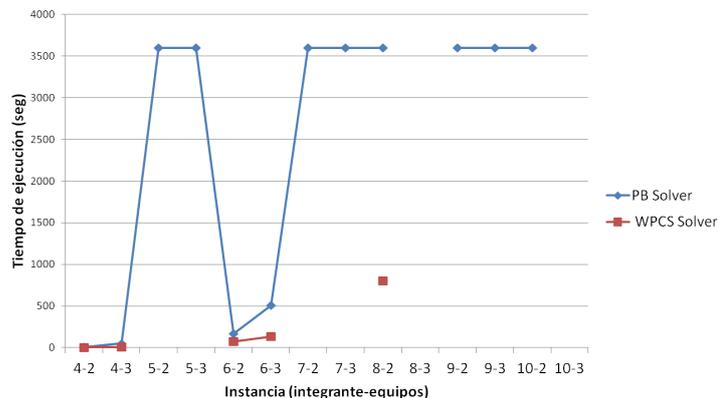


Figura 3. Distribución del tiempo de ejecución (en segundos) según la instancia

bió plasmar en el modelado de las restricciones; y si bien no se hizo un análisis exhaustivo del tratamiento de las restricciones en [9], se supone que la intolerancia a este tipo de escenarios (observar en la Figura 3 que no fueron resueltos) se debe al manejo de ésta en las restricciones.

Por el lado de la segunda prueba, los casos de estudios contaron con la siguiente configuración: todas las restricciones del modelo se definieron como *restricciones débiles* (con lo cual todas formaron parte de la penalización); y se formaron 3, 4 y 5 equipos de 8 integrantes cada uno, así como 3 y 4 equipos de 10 individuos. Además, el tiempo máximo de ejecución se estableció en 1, 2 y 4 horas. Por último, hay que destacar que para este experimento sólo se consideró la última solución encontrada por el algoritmo al momento de finalizar su ejecución.

Resulta interesante observar como varían las soluciones a medida que se amplió el tiempo de ejecución. Al tomar como referencia la penalización global, tal como se puede apreciar en los gráficos de la Figura 4, la relación de ésta con el tiempo no fue inversamente proporcional como se hubiese imaginado, sino que se dio una distribución casi azarosa. Para comprender la causa es necesario analizar cada uno de los equipos del conjunto.

Para ello, se tomó cada una de las instancias (por ejemplo, 3 equipos de 8 integrantes) y se comparó cada uno de los equipos obtenidos para los diferentes tiempos de ejecución (3600 seg, 7200 seg y 14400 seg), tal como se puede ver en la Figura 5. Se observó que, en general, para los tiempos de ejecución “3600” y “7200” ocurría que se establecía un equipo con una penalización muy baja respecto de los demás, es decir que cumplía con la mayoría de las restricciones, pero los otros equipos tendían a una penalización alta, siendo más bien mediocres. Sin embargo, para el tiempo de ejecución “14400”, en la mayoría de los casos, se alcanzaron equipos con una penalización equilibrada, donde todos cumplían un conjunto análogo de restricciones.

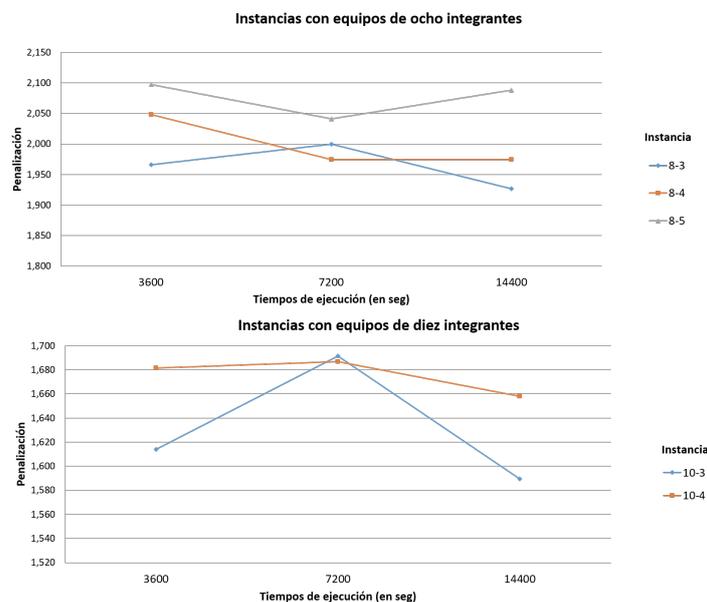


Figura 4. Dispersión de la penalización según los tiempos de ejecución

También es cierto que lo anterior no se cumplió para el caso de la formación de 5 equipos de 8 integrantes, donde la solución encontrada luego de 2 horas (7200 seg) resultó poseer equipos más equilibrados que la alcanzada en 4 horas (14400 seg). Básicamente, una de las causas fue que la complejidad del modelo se disparó respecto a los casos anteriores, con lo cual, se necesitaría darle un mayor tiempo ejecución para alcanzar una solución cuyos equipos tengan una penalización similar.

En resumen, si bien los experimentos realizados son preliminares y están basados en un único dataset, puede decirse que se ha alcanzado uno de los objetivos principales del trabajo: la flexibilidad del modelo. Tal como muestra la segunda prueba, la definición de un modelo cuyas restricciones sean débiles, permite explorar soluciones que no podrían obtenerse con la configuración del primer experimento y el dataset con el que se trabajó. De esta manera, se puede observar una mayor independencia respecto del conjunto de datos por parte de nuestro asistente. Por otro lado, también es válido observar el costo que tiene asociado este tipo de configuraciones: para obtener un resultado estable en la mayoría de los casos hubo que dejar trabajando al asistente durante 4 horas.

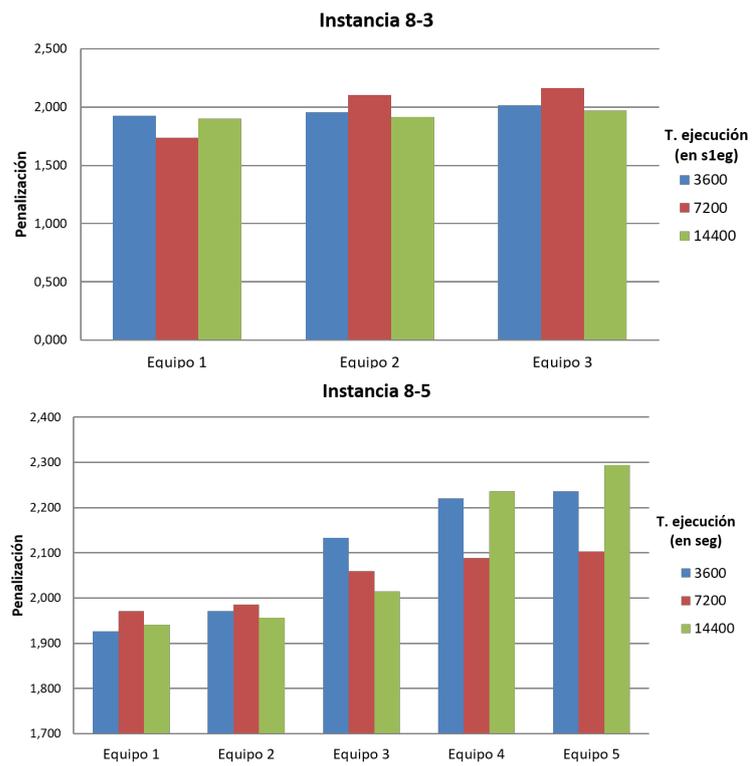


Figura 5. Distribución de la penalización por equipos

5. Conclusiones

En este trabajo se presentó un asistente automatizado basado en un enfoque de satisfacción de restricciones, que considera cuatro características –el perfil psicológico, los roles a desempeñar, las redes sociales y las competencias técnicas–, para la formación de equipos de trabajo. El principal objetivo de éste, fue proponer un enfoque que pudiese lidiar con modelos flexibles.

La utilización del enfoque de optimización pseudo-booleano ofreció la posibilidad de definir modelos donde se conjugaban distintos tipos de restricciones. Esto permitió que sea el seleccionador quien las defina, teniendo la opción de relajar ciertas condiciones. Con esto, se pudo superar la visión de modelo único propuesto en [9] y, a su vez, se logró alcanzar cierto grado de independencia respecto de los datos. En otras palabras, al dejar en las manos del “seleccionador” la definición de las restricciones como *fuertes* o *débiles*, éste puede explorar modelos que ofrezcan soluciones aproximadas. En este caso, algunas condiciones pueden quedar sin cumplirse, lo que implica una penalidad asociada a la solución. Sin embargo, en contraposición a la precisión perdida, el seleccionador puede trabajar sobre modelos que pueden lidiar con las particularidades del conjunto de personas que posee. Asimismo, el seleccionador puede ver cuánto distan los equipos propuestos por el asistente de su modelo ideal, donde todas las condiciones estarían cubiertas (es decir, con costo cero), y que no es posible alcanzar con ese grupo de individuos.

Como trabajos a llevar adelante en un futuro, resulta interesante utilizar otras técnicas del área de Inteligencia Artificial, tal como los algoritmos evolutivos, para poder comparar las soluciones propuestas contra el enfoque pseudo-booleano. Particularmente, los algoritmos evolutivos tienen la capacidad de obtener soluciones aproximadas en tiempos cortos, incluso para instancias complejas. Otro trabajo futuro consiste en validar el enfoque propuesto en otros datasets de usuarios y grupos, especializando el nivel de habilidades técnicas para equipos de desarrollo de software.

Referencias

1. R. M. Belbin. *Managment teams: Way the succeed or fail*. Butterworth-Heineman, Oxford, 1981.
2. R. M. Belbin. *Team roles al work*. Butterworth-Heineman, Oxford, 1993.
3. F. Mumma. *Team work & Team-Roles: What makes your team tick?* HRDQ, 3 edition, 2005.
4. I. Myers. *Manual: The Myers-Briggs Type Indicator*. Educational Testing Service, Princeton, 1962.
5. I. Myers Briggs and P.B. Myers. *Gifts Differing: Understanding Personality Type*. Davies-Black, 1980.
6. R. Mitchell, S. Nicholas, and B. Boyle. The role of openness to cognitive diversity and group processes in knowledge creation. *Small Group Research*, 5(40):535–554, 2009.

7. V. Pieterse, D. G. Kourie, and I. P. Sonnekus. Software engineering team diversity and performance. In *SAICSIT '06*, pages 180–186, South African, 2006.
8. M. F. Winter. *Developing a group model for student software engineering teams*. PhD thesis, University of Saskatchewan, Canada, 2004.
9. J. Bais, J. Giletto, J. Balmaceda, S. Schiaffino, and A. Díaz Pace. An assistant for group formation in cscl based on constraint satisfaction. *Revista Iberoamericana de Inteligencia Artificial*, 17(53):35–45, 2014.
10. K. Anderson. Group formation with a network constraint. Technical report, Carnegie Mellon University, 2008.
11. J. Petke and P. Jeavons. Local consistency and sat-solver. *Journal of Artificial Intelligence Research*, 463:329–351, 2012.

Detección y cierre de ciclos en sistemas SLAM basados en visión estéreo

Gastón Castro, Pablo De Cristóforis y Taihú Pire

Departamento de Computación - Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Resumen El presente trabajo propone una solución al problema de la detección y cierre de ciclos en sistemas de SLAM que utilizan cámaras estéreo como sensor principal. El método propuesto se encuentra dividido en tres etapas: detección de ciclos en la trayectoria, cálculo del desvío cometido en la localización y corrección tanto de la localización actual del robot como del mapa construido hasta el momento. Durante la etapa de detección se utiliza la técnica de *bag-of-words* para calcular la similitud entre imágenes. Para el cálculo de la transformación de cierre y validación de ciclos se utilizan métodos PnP. Finalmente para la corrección de la trayectoria estimada y del mapa reconstruido se utilizan técnicas de optimización de grafos y algoritmos de minimización no lineal.

La solución propuesta se implementó como un módulo para ser incorporado al sistema SLAM S-PTAM (*Stereo Parallel Tracking And Mapping*) de código abierto. Los experimentos realizados con *datasets* de dominio público bajo el *framework* ROS (*Robot Operating System*) muestran que el método de detección y cierre de ciclos mejora drásticamente la estimación de la localización del robot en el entorno y la calidad del mapa construido, permitiendo al mismo tiempo la ejecución del sistema en tiempo real.

Keywords: Loop Closure, Visual SLAM, Stereo Vision, Robotic Vision

1. Introducción

SLAM (*Simultaneous Localization and Mapping*) plantea la construcción de una representación o mapa del entorno mientras se estima, de manera simultánea, la localización del robot en el mismo [1,2]. El mapa se construye estimando las posiciones de marcas del ambiente, las cuales van siendo ajustadas a medida que son observadas desde distintas posiciones. Al utilizar cámaras como sensores y características visuales (*image features*) como marcas del ambiente, el problema se denomina Visual SLAM (VSLAM). Los métodos que resuelven VSLAM deben tratar con la acumulación del error en la estimación de la posición y orientación del robot, que va creciendo de forma no acotada a medida que aumenta la longitud de la trayectoria recorrida y la dimensión del mapa construido. Para abordar este problema los métodos de VSLAM deben contar con la capacidad de detectar una región previamente visitada por el robot y ajustar tanto, la localización del robot como el mapa construido minimizando el error acumulado

hasta el momento. Este problema se conoce como *Loop Closure*, que incluye la detección y cierre de ciclos en la trayectoria realizada por el robot.

En la literatura existen numerosos métodos para la detección y cierre de ciclos, en estos prima el uso de la técnica *bag-of-words* [3]. En [4,5] se presenta FAB-MAP, el cual plantea un modelo probabilístico de complejidad lineal donde se toma en cuenta la dependencia entre las palabras encontradas en las imágenes. En [6,7] se introduce por primera vez un detector de ciclos basado en descriptores binarios para el manejo de vectores *bag-of-words*. Este método, a diferencia de FAB-MAP, no utiliza un modelo probabilístico para tomar en cuenta la similitud de lugares previamente visitados. En vez de esto, introducen el concepto de consistencia temporal (*temporal consistency*) de manera que grupos de imágenes tomadas en momentos cercanos se consideran de manera conjunta aumentando la confiabilidad de la detección. Por otro lado, al utilizar descriptores binarios el computo de vectores *bag-of-words* es muy veloz, así como también la comparación entre estos. En [8] se utiliza la biblioteca DBoW [7] para la implementación de un sistema monocular capaz de detectar ciclos a nivel de solo algunos cuadros clave (*keyframes*) provistos por la cámara.

El presente trabajo propone una solución al problema de la detección y cierre de ciclos en sistemas VSLAM que utilizan cámaras estéreo como sensor principal. Para lograr este objetivo se divide el problema en tres etapas: la detección de ciclos en la trayectoria, el cálculo del desvío cometido en la localización y la corrección tanto de la posición y orientación actual del robot como del mapa construido hasta el momento.

2. S-PTAM

Pire et al.[9] presenta un sistema SLAM basado en visión estéreo, capaz de estimar la localización de un robot móvil en tiempo real en trayectorias de gran longitud. S-PTAM se inspira en PTAM [10] aprovechando la capacidad de cómputo de unidades de procesamiento paralelo, dividiendo el problema al igual que PTAM en dos tareas principales: seguimiento de la cámara (*Tracking*) y la construcción del mapa (*Local Mapping*).

Al inicio del sistema, S-PTAM considera la pose de la cámara en el centro de coordenadas del mundo, y un mapa que será inicializado a partir de la triangulación de las características visuales presentes en el primer par de imágenes obtenidas de la cámara estéreo. Desde ese momento, el hilo del *Tracking* estima la pose actual de la cámara por cada nuevo par de imágenes, minimizando el error de re-proyección entre las características visuales en las imágenes y puntos existentes en el mapa. Ocasionalmente, algunos *frames* son estratégicamente seleccionados de manera de aumentar el tamaño del mapa, a través de la triangulación de las características visuales extraídas. A estos *frames* seleccionados se los denomina *keyframes* y proveen restricciones geométricas que son utilizadas para el refinamiento de los puntos del mapa. El hilo del *Local Mapping* se encarga de optimizar las poses de los *keyframes* y puntos del mapa recientemente agrega-

dos. Esta optimización tiene como objetivo minimizar el error de re-proyección de manera de incrementar la precisión del mapa.

3. Método de detección y cierre de ciclos propuesto

Se extiende el funcionamiento de S-PTAM añadiendo un módulo de detección y cierre de ciclos denominado *Loop Closing*. Este módulo trabaja de manera paralela en un hilo de ejecución distinto a los del *Tracking* y *Local Mapping*. Los *keyframes* procesados por el *Local Mapping* son luego almacenados para ser tratados por el hilo de *Loop Closing* (Fig. 1). *Loop Closing* procesa los *keyframes* de forma secuencial, realizando la detección, validación y cierre de ciclos. En caso de validar la ocurrencia de un ciclo, la corrección del ciclo y la actualización del mapa se realizan sin entorpecer el funcionamiento general del sistema S-PTAM. En caso de que se requiera corregir la trayectoria actual en curso, el módulo notifica los cambios en la estimación de localización actual del sistema.

En la etapa de detección, los *keyframes* provistos por el *Local Mapping* son convertidos a vectores *bag-of-words* utilizando un vocabulario visual previamente entrenado. En base a los vectores *bag-of-words* obtenidos se construye una base de datos de *keyframes*, como se propone en [8]. Por cada nuevo *keyframe* agregado al mapa, se consulta la base de datos de manera de obtener todo *keyframe* anterior que sea similar en términos de apariencia.

En caso de establecer un candidato a ciclo entre dos *keyframes*, se procede con un análisis de consistencia geométrica que permite validar la ocurrencia de este ciclo. Para esto se utilizan métodos de Perspectiva por n Puntos [11,12] para el cálculo de la transformación relativa entre los *keyframes* involucrados utilizando únicamente las imágenes. Esta transformación permite medir el error acumulado hasta el momento y validar el ciclo detectado.

Una vez que un ciclo es considerado válido, se propaga una corrección entre todos los *keyframes* pertenecientes al ciclo. Esta propagación provee de una solución inicial para una posterior optimización del grafo de poses de los *keyframes*. La optimización ajusta la posición y orientación de los *keyframes* reduciendo el error acumulado en las estimaciones.

3.1. Detección de ciclos

La detección de ciclos se realiza haciendo uso del eficiente método basado en apariencias propuesto por [6]. Cada nuevo *keyframe* K_i agregado al mapa es convertido a un vector *bag-of-words* v_i el cual es utilizado para consultar la base de datos de *keyframes* de manera de obtener todo v_j anterior que comparta palabras con v_i . La similitud entre vectores *bag-of-words* se evalúa a través del valor de similitud normalizado (*normalized similarity score*):

$$\eta(v_i, v_j) = \frac{s(v_i, v_j)}{s(v_i, v_{i-1})}$$

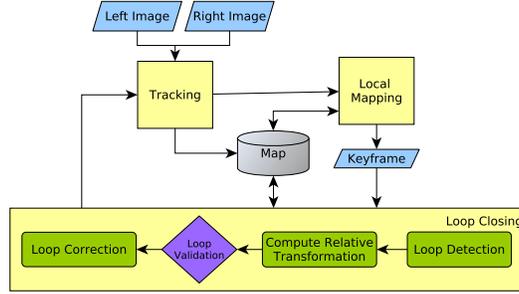


Figura 1. Esquema general del sistema S-PTAM incluyendo detección y cierre de ciclos.

donde $v_{i-1>0}$ es el vector *bag-of-words* asociado al *keyframe* anterior a K_i y el valor de similitud entre dos vectores *bag-of-words* $s(v_i, v_j)$ corresponde a la evaluación L_1 ($L_1 - score$) propuesta por [13]:

$$s(v_i, v_j) = 1 - \frac{1}{2} \left| \frac{v_i}{|v_i|} - \frac{v_j}{|v_j|} \right|$$

donde $|v|$ corresponde a la norma L_1 del vector v y $s(v_i, v_j) \in [0 \dots 1]$. Se establece un *threshold* mínimo permisivo de ($\alpha = 0,3$) sobre el valor de similitud normalizado dado por $\eta(v_i, v_j)$, considerando como candidatos a ciclos aquellos *keyframes* pertenecientes a la base de datos que superen dicho *threshold*. En caso de que la detección de ciclos en la trayectoria sea positiva, es caracterizada como un par de *keyframes* K_c, K_ℓ pertenecientes al mapa. K_c referencia al *keyframe* recientemente procesado que originó la consulta y K_ℓ al mejor candidato a ciclo obtenido de la base de datos.

3.2. Validación del ciclo y cálculo de la transformación relativa

Habiendo establecido un par de *keyframes* K_c, K_ℓ como un candidato a ciclo en la trayectoria, se lleva a cabo un análisis geométrico para validar la ocurrencia del mismo. Concretamente, se analiza si es posible establecer una transformación relativa $T_{c\ell}$ entre las poses de cámara de K_c, K_ℓ utilizando técnicas de Perspectiva por n Puntos [14].

Para llevar a cabo esta validación, primero es necesario comparar y asociar los descriptores extraídos de K_c y K_ℓ de manera de establecer correspondencias entre las características visuales detectadas en las imágenes. Sobre esta correspondencia entre descriptores se utiliza el método P3P propuesto por [11] bajo un esquema RANSAC de manera de determinar una transformación entre los *keyframes* K_c, K_ℓ consistente con una cantidad considerable de correspondencias *inliers*. En caso de que al menos un 80% del total de correspondencias sean encontradas *inliers*, el ciclo entre los *keyframes* K_c, K_ℓ se considera válido. Finalmente se estima la transformación relativa $T_{c\ell}$ entre las poses de cámara de los

keyframes K_c, K_ℓ utilizando el método UPnP [12] únicamente sobre el conjunto de correspondencias consideradas como *inliers*.

3.3. Cierre del ciclo y corrección del mapa

El proceso de cierre y corrección del ciclo estima en una primera etapa, una corrección inicial propagando la transformación $T_{c\ell}$ por los *keyframes* comprendidos entre K_ℓ y K_c .

Esto es, sean $\{E_{w_c}, \dots, E_{w_{(j+1)}}, E_{w_j}, \dots, E_{w_\ell}\}$ las poses de cámara asociadas a los *keyframes* pertenecientes al ciclo detectado, la propagación se define como:

$$\begin{aligned} E_{w_c}^{prop} &= E_{w_\ell}(T_{c\ell})^{-1} = E_{w_\ell}T_{\ell c} \\ E_{w_j}^{prop} &= \text{Interpolar}_j(E_{w_j}, E_{w_{(j+1)}}^{prop} E_{(j+1)j}) \\ E_{w_\ell}^{prop} &= E_{w_\ell}, \end{aligned}$$

donde el índice superior “*prop*” refiere a las poses de cámara luego de la propagación, $T_{\ell c} = (T_{c\ell})^{-1}$ corresponde a la transformación inversa de $T_{c\ell}$ previamente estimada y $E_{(j+1)j}$ es la transformación entre las poses $E_{w_{(j+1)}}$ y E_{w_j} ($E_{(j+1)j} = (E_{w_{(j+1)}})^{-1}E_{w_j} = E_{(j+1)w}E_{w_j}$). $\text{Interpolar}_j(\bullet, \bullet)$ realiza una interpolación entre poses de manera de “suavizar” la propagación efectuada sobre el *keyframe* K_j , de acuerdo a su distancia al ciclo detectado. De esta manera, *keyframes* cercanos a K_c (y por tanto cercanos al punto donde fue detectado el ciclo) serán fuertemente corregidos, mientras que la corrección aplicada a *keyframes* alejados del punto de cierre de ciclo es suavizada. Esta corrección inicial logra cerrar el ciclo “uniendo” a K_c y K_ℓ a través de $T_{c\ell}$, propagando una corrección suavizada a lo largo de todos los *keyframes* involucrados en el ciclo.

Luego de la corrección inicial, se lleva a cabo una optimización del grafo de poses formado por los *keyframes* del mapa. Esta optimización ajusta la posición y orientación de los *keyframes* de manera de que satisfagan las restricciones geométricas previamente establecidas entre los *keyframes*. En términos de minimización, se define el error residual entre poses de cámara con respecto a las restricciones de transformación. De esta manera, a través de algoritmos de minimización, se obtienen las poses de cámara que minimicen el error acumulado de todos los errores residuales. Sea $E_{i(i+1)}, \forall i \in [0, \mathcal{C}-1]$, la transformación entre la pose de cámara E_{w_i} y la subsecuente pose $E_{w_{(i+1)}}$. Se define como el error residual entre *keyframes* subsiguientes como $r_i = E_{i(i+1)}(E_{w_{(i+1)}})^{-1}E_{w_i} = E_{i(i+1)}E_{(i+1)w}E_{w_i}$. Para todo par de *keyframes* K_j, K_k tal que se haya validado un ciclo entre estos, se define el error residual entre *keyframes* que cierran ciclos como $r_{j,k} = T_{jk}(E_{w_k})^{-1}E_{w_j} = T_{jk}E_{kw}E_{w_j}$. Las transformaciones T_{jk} corresponden a las transformaciones estimadas en la etapa de validación de cada ciclo. De esta manera interesa ajustar las poses de cámara, de manera de minimizar los errores residuales definidos:

$$\underset{E_{w_0}, \dots, E_{w_c}}{\text{argmin}} \sum_i r_i^\top r_i + \sum_{j,k} r_{j,k}^\top r_{j,k}$$

La solución al cierre del ciclo propagada inicialmente sirve como estimación inicial para los métodos de minimización no lineal utilizados [15,16], reduciendo considerablemente el tiempo requerido para optimizar los *keyframes* del mapa. Finalmente, una vez ajustadas las poses de los *keyframes*, cada punto del mapa es corregido aplicando la misma transformación que haya sufrido el *keyframe* que lo trianguló originalmente.

3.4. Actualización del mapa y sincronización de componentes

La actualización de los keyframes y puntos pertenecientes al mapa es un proceso de alto costo computacional. Para permitir al sistema operar en tiempo real junto con la extensión de detección y cierre de ciclos, se dividió la actualización del mapa en dos etapas. En una primera etapa, se actualiza el área del mapa que no esta siendo activamente utilizada por los hilos del *Tracking* y *Local Mapping*. Esta área corresponde a la mayor parte del mapa y se actualiza de manera concurrente sin requerir la detención de dichos hilos. En una segunda etapa, se actualiza el área del mapa que se encuentra activa. Esta área corresponde a una porción reducida del mapa por lo que su actualización resulta veloz, requiriendo la detención del *Tracking* y *Local Mapping* por solo un breve periodo de tiempo.

4. Experimentación

A partir de la solución propuesta para la resolución del problema planteado, se diseñaron diferentes experimentos que permitieron evaluar la precisión, robustez y eficiencia de todos los métodos involucrados en la detección y cierre de ciclos.

En todos los experimentos se utiliza el detector de esquinas Shi-Tomasi [17] y el descriptor binario BRIEF [18] para la detección y extracción de características visuales de las imágenes. El hardware utilizado para el procesamiento de los experimentos corresponde a una computadora estándar de escritorio Intel(R) Core(TM) i7-860 2.80GHz y 16GB RAM.

4.1. The KITTI dataset

The KITTI Vision Benchmark Suite [19] es una colección de *datasets* capturados por un vehículo transitando por las calles de la ciudad de Karlsruhe, Alemania. El *dataset* consta de imágenes estéreo con una resolución de 1344×391 *pixeles* a una frecuencia de 10 *frames* por segundo (10Hz). La colección se compone de 10 secuencias que proporcionan información de la trayectoria real (*ground truth*). Asimismo, ocurren ciclos en 6 de estas secuencias (las secuencias: 00, 02, 05, 06, 07 y 09). Se utiliza el *ground truth* sobre la ocurrencia de ciclos propuesto por Arroyo et al.[20] para evaluar el desempeño de la detección y validación de ciclos.

Por otro lado, el *dataset* presenta un *ranking* comparativo entre diferentes métodos SLAM del estado del arte¹ que permite comparar el desempeño del sistema S-PTAM con la inclusión del módulo de detección y cierre de ciclos.

Precisión en la detección y cantidad de ciclos reconocidos

Dado el comportamiento de selección de *keyframes* propio de S-PTAM, existen cuadros que no son agregados al mapa y no son procesados por el *Loop Closing*. Por esta razón se analizan dos medidas de reconocimiento de ciclos, una sobre el total de ciclos de la secuencia (%Recon.) y una sobre aquellos ciclos que fueron observados dada la selección de *keyframes* realizada durante la ejecución del sistema (%Recon. Obs.).

La tabla 1 muestra los resultados obtenidos por el método de detección de ciclos diferenciando el desempeño del análisis de apariencia y el de la validación geométrica. El análisis de apariencia resulta muy permisivo generando gran cantidad de candidatos a ciclos y un porcentaje de reconocimiento satisfactorio tomando en cuenta otros métodos del estado del arte [20]. La validación geométrica descarta falsos positivos de manera efectiva logrando una precisión del 100% en las detecciones validadas. La cantidad de ciclos finalmente validados es proporcionalmente bajo en comparación a los ciclos definidos por el *ground truth*. Esto no resulta contraproducente dado que, al requerir un porcentaje elevado de *inliers*, los ciclos validados permiten la estimación de transformaciones relativas más precisas. De esta manera se prioriza la validación de una menor cantidad de ciclos, pero que resulten en estimaciones de mayor precisión.

Cuadro 1. Resultados de precisión y reconocimiento en las diferentes secuencias.

Secuencia	#Ciclos	Apariencia				Apariencia + Validación Geométrica			
		#Detec.	%Precisión	%Recon.	%Recon. Obs.	#Valid.	%Precisión	%Recon.	%Recon. Obs.
KITTI00	732	2747	13,14 %	49,32 %	57,39 %	45	100 %	6,14 %	7,15 %
KITTI02	234	3010	3,12 %	40,17 %	40,52 %	5	100 %	2,14 %	2,16 %
KITTI05	320	1596	12,4 %	61,88 %	68,51 %	28	100 %	8,75 %	9,69 %
KITTI06	269	412	24,03 %	36,8 %	38,98 %	16	100 %	5,95 %	6,30 %
KITTI07	13	434	2,07 %	69,23 %	100 %	1	100 %	7,69 %	11,11 %
KITTI09	17	836	0,60 %	29,41 %	35,71 %	0	/	0 %	0 %

La Figura 2(b) muestra las detecciones validadas en la secuencia 00, se expone la trayectoria *ground truth* en función del tiempo de manera de visualizar las asociaciones realizadas por el método (líneas rojas). Es posible observar que si bien la validación geométrica reduce considerablemente la cantidad de ciclos reconocidos, valida al menos uno en cada segmento previamente visitado de la trayectoria. Este comportamiento es favorable ya que reduce la cantidad de cierres de ciclos realizados por el sistema y concentra el procesamiento en ciclos con mayor cantidad de *inliers* donde las transformaciones relativas estimadas son más precisas.

¹ http://www.cvlibs.net/datasets/kitti/eval_odometry.php

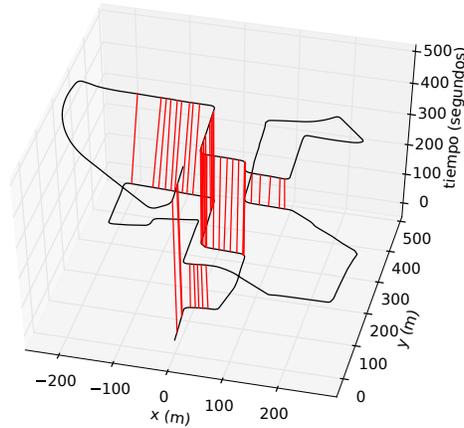


Figura 2. Ciclos detectados y validados por el método en la secuencia KITTI 00.

Trayectoria estimada y reducción del error acumulado

Se exponen los resultados obtenidos durante la ejecución del sistema en la secuencia 00. Se presenta una comparación de la trayectoria estimada por el sistema al incluir la extensión de *Loop Closing* (Fig. 3(a)), el resultado final del mapa luego de la corrección de los ciclos (Fig. 3(b)) y un análisis “momento a momento” de los errores de traslación y rotación cometidos en la localización (Figs. 4(a) y 4(b)).

Es posible observar frecuentes oscilaciones en el error de traslación debido a lo que se conoce como “cancelación del error”. El error acumulado en alguna de las direcciones puede verse cancelado posteriormente por la acumulación de error en el sentido contrario. Por el contrario, la calidad del mapa construido decrece progresivamente potenciando la acumulación de error en estimaciones posteriores. Se puede observar una disminución del error absoluto de traslación de manera consistente desde el momento en que se produce la primera detección (Fig. 4(a)). El error angular presenta un comportamiento errático, sin embargo se puede apreciar una mejora en la estimación. Observar que la corrección de sucesivos ciclos temporalmente cercanos no se traduce en una mayor reducción del error de estimación. Dado que el sistema acumula error en intervalos de tiempo donde no ocurren ciclos, el mayor impacto en la precisión lo produce el primer ciclo corregido luego de un intervalo de tiempo prolongado donde no ocurriera ninguno. Asimismo, el error en la trayectoria estimada se encuentra dentro del rango de errores obtenidos por los métodos del estado del arte publicados en la página del *dataset*.

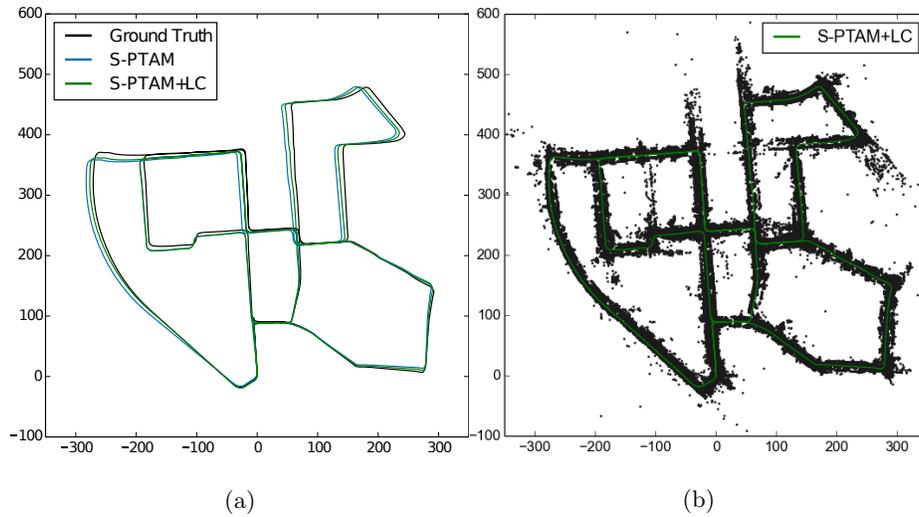


Figura 3. Resultados del método en la secuencia KITTI 00. (a) Trayectoria estimada por el sistema S-PTAM original y al incluir la extensión de *Loop Closing*. (b) Visualización del mapa construido y corregido por el método luego de los cierres de ciclos.

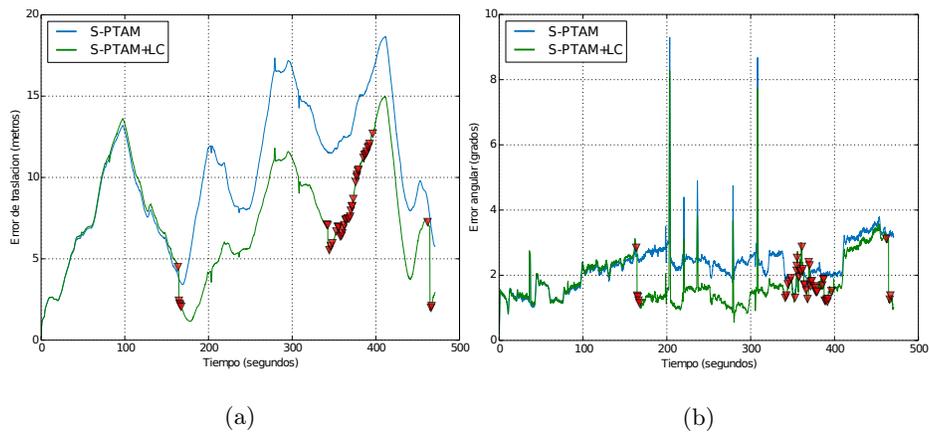


Figura 4. Resultados del método en la secuencia KITTI 00. Error de traslación y angular cometido al estimar la localización en cada momento de la secuencia. Los marcadores rojos indican los momentos en que se efectuaron correcciones de ciclo.

4.2. QUT Level 7 S-Block dataset

El *Level 7 S-Block dataset* [21] provee información recolectada durante el recorrido realizado por un robot terrestre en el nivel 7 del *S-Block QUT Gardens Point campus* en Brisbane, Australia. El robot provee 12 imágenes estéreo por segundo (12Hz) y estimaciones odométricas de posición basadas en la rotación de las ruedas. La secuencia dura aproximadamente 33 minutos y el recorrido realizado corresponde a una serie de vueltas en un ambiente interior (*indoor*) de oficinas. No se cuenta con información de *ground truth* sobre los ciclos de la trayectoria, pero estos ocurren frecuentemente permitiendo comprobar la estabilidad del cierre y medir los tiempos de procesamiento requeridos.

Trayectoria estimada y reducción del error acumulado

En este *dataset* las estimaciones generadas por el S-PTAM original presentan una acentuada acumulación de error (Fig. 5(a)). El desempeño del sistema se degrada progresivamente manifestando un patrón oscilatorio que crece en amplitud. Esta degradación llega al punto de registrar errores de 25 metros en la traslación y 90 grados en la orientación (Figs. 6(a) y 6(b)). Esto imposibilita la utilización del sistema S-PTAM original como sistema de localización en este escenario.

La inclusión del módulo de *Loop Closing* mejora notoriamente el desempeño exhibiendo errores máximos de 2,2 metros en la traslación y 14 grados en la orientación. El sistema es capaz de realizar correctas asociaciones de apariencia a pesar de tratarse de un entorno repetitivo, en total se validan y cierran 317 ciclos (Fig. 5(b)). Se puede apreciar que la amplitud de los errores registrados se encuentra acotada y no crece con el tiempo a diferencia del comportamiento observado en el S-PTAM original (Figs. 6(a) y 6(b)).

4.3. Tiempos de procesamiento

Durante la secuencia se realizan 3706 análisis de apariencia, 2218 análisis geométricos y 317 cierres de ciclos. La Tabla 2 presenta los tiempos de procesamiento consumidos por cada una de las etapas del módulo *Loop Closing*. La actualización del área compartida del mapa resulta muy veloz requiriendo la detención del *Tracking* y el *Local Mapping* por 1,6 milisegundos en promedio.

Cuadro 2. Tiempos de procesamiento consumido por las diferentes etapas.

Etapas	#Mediciones	Mín.(ms)	Prom.(ms)	Máx.(ms)
Detección	3706	0	0,74	8
Val. y transf. relativa	2218	0	2,31	11
Cierre del ciclo	317	42	384,92	922
Actualización de mapa	317	17	80,53	187
Actualización de mapa (área de utilización compartida)	317	0	1,6	4

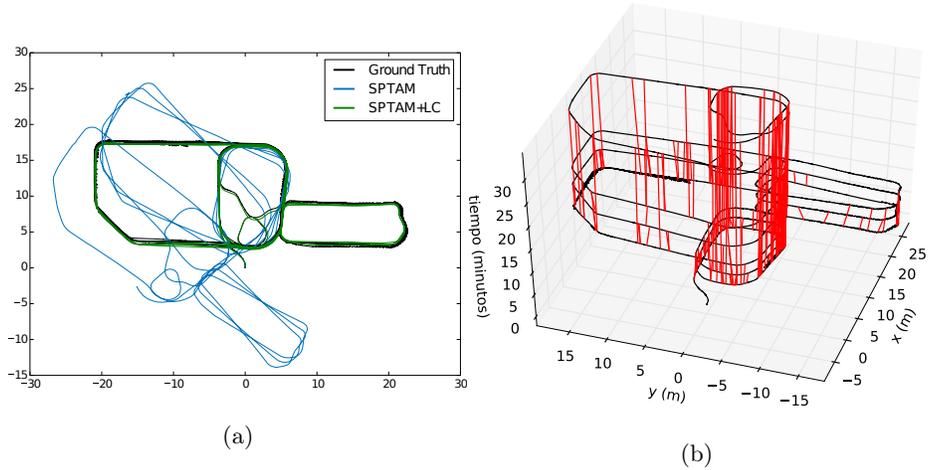


Figura 5. Resultados del método en el *Level7 dataset*. (a) Trayectoria estimada por el sistema S-PTAM original y al incluir la extensión de *Loop Closing*. (b) Ciclos detectados y validados, la trayectoria se encuentra representada en función del tiempo.

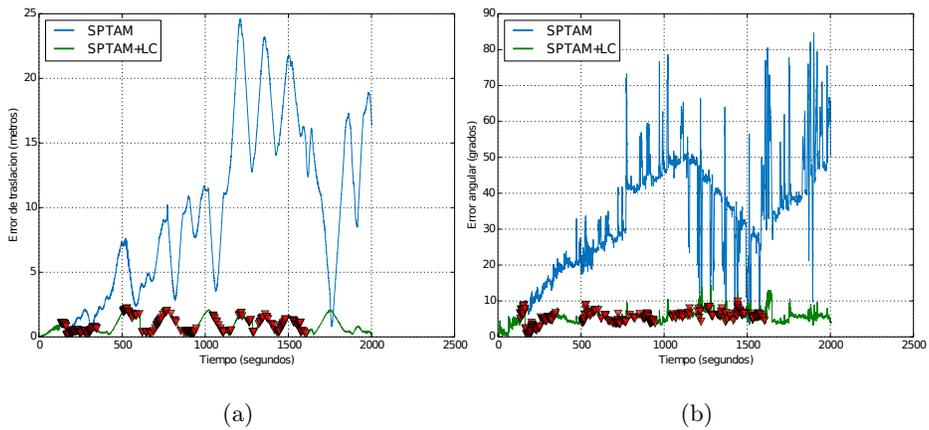


Figura 6. Resultados del método en el *Level7 dataset*. Error de traslación y angular cometido al estimar la localización en cada momento de la secuencia. Los marcadores rojos indican los momentos en que se efectuaron correcciones de ciclo.

5. Conclusiones

En este trabajo se presenta el desarrollo de un método de detección y cierre de ciclos integrado a un sistema SLAM basado en visión estéreo capaz de trabajar en tiempo real. En particular, el desarrollo e integración se realizó sobre el sistema SLAM S-PTAM propuesto en [9]. La evaluación se llevó a cabo utilizando *datasets* de dominio público permitiendo la comparación de los resultados obtenidos con el estado del arte. Los resultados obtenidos muestran que el método de detección y cierre de ciclos reduce y mantiene acotado el error global acumulado por el sistema de SLAM estéreo S-PTAM, sin impactar negativamente en su rendimiento. Tanto los errores de traslación como los errores de rotación relativos y globales obtenidos se encuentran dentro del rango de errores reportados en el estado del arte de SLAM.

Como trabajo futuro se propone reutilizar los módulos de detección y validación del método propuesto para resolver el problema de relocalización. Por otro lado, se planea incorporar un método de fusión de puntos que permita fusionar aquellos puntos que hagan referencia a una misma marca visual. De esta manera no solo se mantiene una reconstrucción del mapa más precisa, y por ende una mejor localización, sino que también se reduce el consumo de memoria del sistema.

Referencias

1. H. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics Automation Magazine, IEEE*, 13(2):99–110, June 2006.
2. Tim Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (slam): part ii. *Robotics Automation Magazine, IEEE*, 13(3):108–117, Sept 2006.
3. Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.
4. Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.
5. Mark Cummins and Paul Newman. Accelerating fab-map with concentration inequalities. *Robotics, IEEE Transactions on*, 26(6):1042–1050, 2010.
6. Dorian Galvez-Lopez and Juan D Tardos. Real-time loop detection with bags of binary words. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 51–58. IEEE, 2011.
7. Dorian Galvez-Lopez and J. D. Tardos. Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197, October 2012.
8. Raúl Mur-Artal and Juan D Tardós. Fast relocalisation and loop closing in keyframe-based slam. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 846–853. IEEE, 2014.
9. Taihú Pire, Thomas Fischer, Javier Civera, Pablo De Cristóforis, and Julio Jacobo Berlles. Stereo parallel tracking and mapping for robot localization. In *International Conference on Intelligent Robots and Systems*, Hamburg, Germany, September 2015.

10. Georg Klein and David Murray. Parallel Tracking and Mapping for Small AR Workspaces. In *ISMAR*, pages 1–10, Washington, DC, USA, 2007. IEEE Computer Society.
11. Laurent Kneip, Davide Scaramuzza, and Roland Siegwart. A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2969–2976. IEEE, 2011.
12. Laurent Kneip, Hongdong Li, and Yongduek Seo. Upnp: An optimal $O(n)$ solution to the absolute pose problem with universal applicability. In *Computer Vision–ECCV 2014*, pages 127–142. Springer, 2014.
13. David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. IEEE, 2006.
14. Bert M Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *International journal of computer vision*, 13(3):331–356, 1994.
15. Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. 1944.
16. Donald W Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the society for Industrial and Applied Mathematics*, 11(2):431–441, 1963.
17. J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pages 593–600, Jun 1994.
18. M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision*, pages 778–792, 2010.
19. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, page 0278364913491297, 2013.
20. Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, J Javier Yebes, and Sebastián Bronte. Fast and effective visual place recognition using binary codes and disparity information. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 3089–3094. IEEE, 2014.
21. Liz Murphy, Timothy Morris, Ugo Fabrizio, Michael Warren, Michael Milford, Ben Upcroft, Michael Bosse, and Peter Corke. Experimental comparison of odometry approaches. In *Experimental Robotics*, pages 877–890. Springer, 2013.

Índice de Autores

Aloi, Federico	5
Castro, Gastón Ignacio	32
De Cristóforis, Pablo	32
Gatti, Ignacio	19
Gomez Fernandez, Francisco	3
Gálvez, Ramiro H.	1
Heredia Favieri, Nadia Mariel	3
Pire, Taihú	32
Pérez, Juan Manuel	7

Índice de Palabras Clave

análisis estático de código	5
aprendizaje automático	1
automatización de corrección de programas	5
Cierre de ciclos	32
descriptor	3
Deteccion de ciclos	32
educación	5
enseñanza de la programación	5
Entrainment	7
Inteligencia Artificial	19
object recognition	3
Optimización en formación de equipos	19
point cloud	3
predicción de retorno de acciones	1
Procesamiento del Habla	7
procesamiento del lenguaje natural	1
programación	5
RGB-D	3
Robótica móvil	32
Series de Tiempo	7
Técnicas SAT	19
Visión estéreo	32
Visual SLAM	32